ATARI

ST COMPUTER

Die Fachzeitschrift für den ATARI-ST Anwender.

März '86
ÖS 53,—/Sfr. 6,—
DM 6,—
NR. 3

* Zwei Fremdfloppies am ST
* Messebericht Micro '86
* Bericht über First Word, DB Master One
* GEM-Routinen praktisch genutzt
* 68 000er Prozessor unter der Lupe

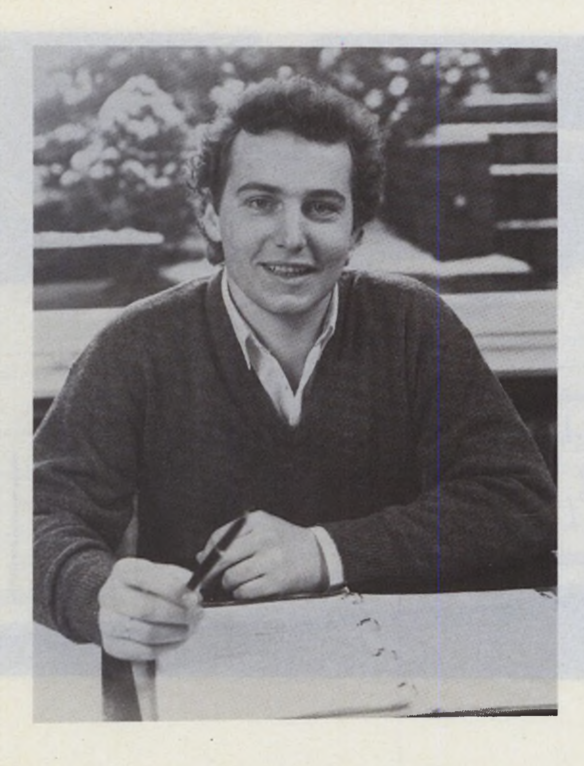
Umsatz 1.Quartal 1986

ATARI

ATARI 520ST

Test: Business-Software

VIP Professional



Liebe ST Leser!

An dieser Stelle möchte ich mich zuerst einmal ganz herzlich bei den vielen Lesern bedanken, die uns – die Redaktion – durch Ihre negative wie positive Kritik unterstützt haben. Denn gerade in einer immer etwas turbulenten Anfangsphase ist eine Resonanz der Leser von großer Bedeutung. Daher freuen wir uns wirklich über jeden Brief, egal, ob er nur Fragen oder Kritik enthält, oder Sie uns einfach Ihre Meinung über ein Thema mitteilen. Die ST Computer ist eine Fachzeitschrift von Anwendern für Anwender und dazu benötigen wir Ihre Anregungen, um uns ständig neue Ziele zu setzen und damit überhaupt Verbesserungen möglich sind. Doch haben Sie bitte Verständnis dafür, daß wir nicht jeden Brief persönlich beantworten können, denn wir sind bemüht, für Sie immer rechtzeitig viele und gute Informationen zu sammeln und das kostet eine Menge Zeit. Fragen bzw. Berichte die von allgemeinem Interesse sind, werden natürlich veröffentlicht. So wird es ab der nächsten Ausgabe eine Rubrik geben, in der spezielle Leserfragen, kurze Berichte, sowie Tips veröffentlicht werden. Wir hoffen, daß damit ein reger Informationsaustausch auch zwischen den Lesern möglich wird.

Wenn Ihnen also der eine oder andere Artikel mißfällt, so nehmen Sie meine Aufforderung ernst und schreiben uns. Sie werden sehen, daß auch **Sie** an der Gestaltung der ST Computer mitwirken können.

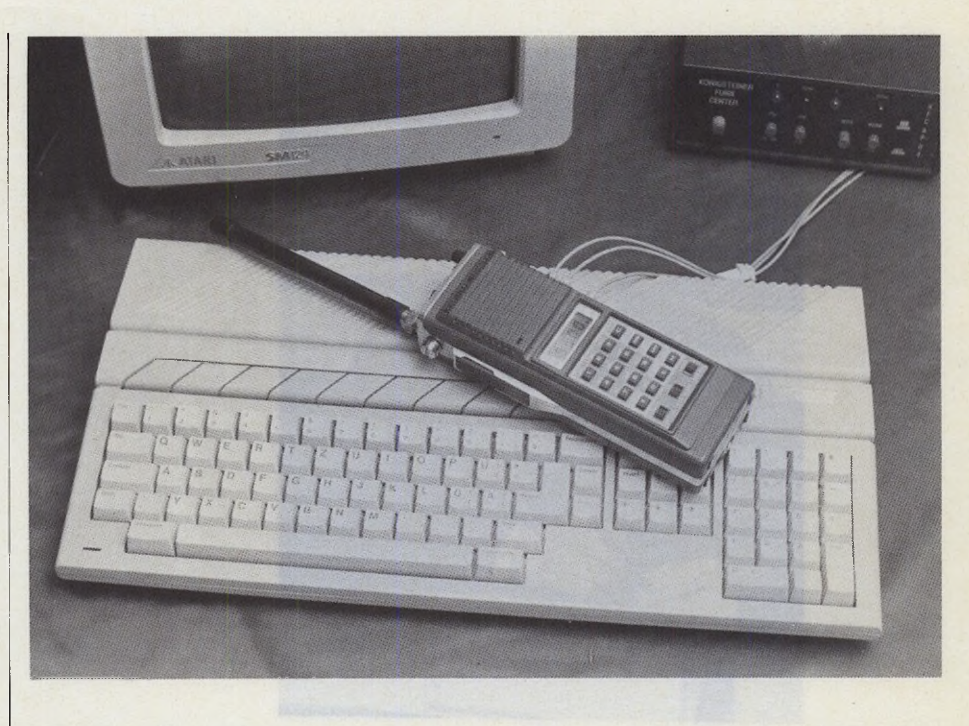
Uwe Bärtels

Inhalt

Allgemeines

Hardware

Software



Messebericht

Listing

GEM

Kurse

CP/M

Aktuelles

File Edit Block Style Help
A:\1STWORD.ORD\AAA.
A:\1STWORD.ORD\TEST.DOC
Betrifft : tangentiale Beschleunigung
Copyright (C) 1985 GST Holdings Limited. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic or mechanical, magnetic, optical, chemical, manual or otherwise, without the prior written permission of GST Holdings



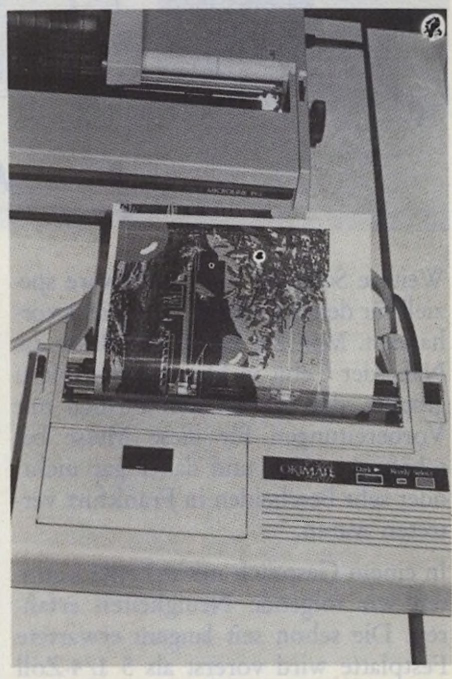
Messebericht Micro-Computer '86

Vom 29.1.–2.2.1986 fand in Frankfurt die zweite „Micro-Computer"-Ausstellung statt. Der Erfolg des Vorjahres ließ große Erwartungen aufkommen. Unter dem Motto „kommen, sehen, testen, verstehen" konnte man die Angebote von über 160 Ausstellern auf ca. 10 000 Quadratmetern Ausstellungsfläche betrachten. Die Messe, die in Zusammenarbeit mit dem Microcomputermagazin CHIP und der Messe Frankfurt veranstaltet wurde, hatte einiges zu bieten. Es gab Schwerpunkte im Gesundheitswesen, Handwerk, Industrie, Bildung und Handel. Außerdem wurden Sonderschau für „Computer und Musik", sowie die Computer- und Software des Jahres 1985 angeboten. Im Jugend-Zentrum konnte Jung und Alt nach Herzenslust Computer verschiedener Hersteller „ausprobieren". Die Programme, die dort zu sehen waren, waren größtenteils Spiele oder Demonstrationsprogramme.

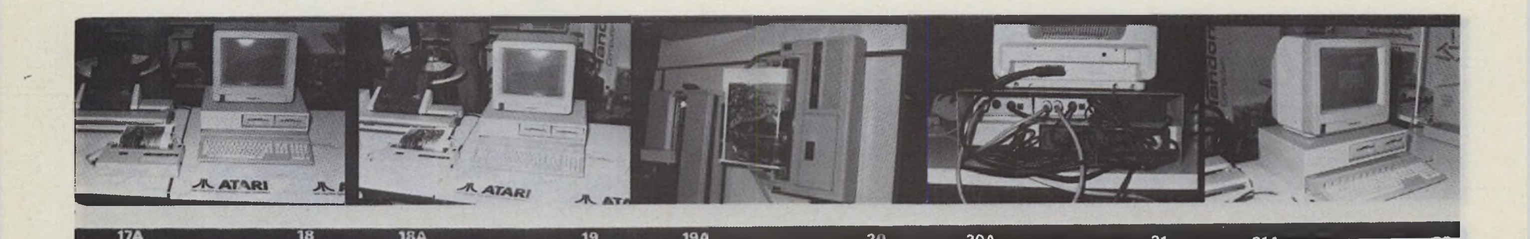
Die Firma ATARI war bei dieser Messe in zwei Hallen vertreten. Einmal am Stand des „Computer des Jahres '85" und noch einmal im „Workshop" Jugend-Zentrum. Nicht viel Neues gab es in Sachen Software für den ATARI ST zu sehen. Außer ein paar neuen gutgelungenen Farbdemos, wie z. B. ein dreidimensionales simuliertes Schachspiel, gab es für berufliche Anwendung nur ein interessantes Softwarepaket. Von der Firma KRAY-ELECTRONIC aus Göttingen wurde ein komplettes Abrechnungssystem für Ärzte vorgestellt. Das System beinhaltet neben der notwendigen Software, einen ATARI 260 ST, ein Laufwerk SF 314, den Monitor SM 124, eine 20 Megabyte Festplatte, einen OKI Drucker Microline 192 und ein Modem zur Datenfernübertragung. Das Ganze wird mit einer Holzkonsole für 9 900,– DM zzgl. MwSt. geliefert. Das Programm, welches in „C" geschrieben ist nutzt die Möglichkeiten des GEMs. Wie uns am Stand gesagt wurde, wird das System den speziellen Erwartungen der Kunden angepaßt.

Am Stand „Software des Jahres '85" konnten wir von einem Vertreter des amerikanischen Software-Entwicklers BORLAND erfahren, daß eine neue TURBO PASCAL Version speziell für die 68 000er Generation entwickelt wird. Diese Version wird dann voraussichtlich Turbo Pascal 4.0 heißen und auch für die ATARI ST Serie erhältlich sein. Das Programm soll noch in der ersten Hälfte 1986 erscheinen.





Das Computer Center Königstein (KFC) zeigte als einziger Anbieter den ATARI ST in einem neuen Gewand. Dieses ähnelt einem PC-Gehäuse (siehe Bild) und besteht aus Stahlblech und bietet Platz für zwei Floppy-Laufwerke und den „Kabelsalat" des ST. Das Gehäuse wird von der Firma



CSF GmbH aus Bielefeld hergestellt. Am gleichen Stand konnte man dem Farbdrucker Okimate 20 von OKIDATA beim Drucken einer farbigen Bildschirmhardcopy des ST zusehen. Die benötigte Treibersoftware für den ATARI ST wird beim Kauf als Sonderleistung mitgeliefert.



Weitere Software oder Hardware speziell für den ATARI ST war nicht vorhanden. Man hat bei dieser Messe die Nähe der Cebit in Hannover insofern gespürt, daß die meisten Firmen mit Vorbereitungen für diese Messe beschäftigt waren, und daher gar nicht, oder sehr bescheiden in Frankfurt vertreten waren.

In einem Gespräch mit ATARI konnten wir folgende Neuigkeiten erfahren. Die schon seit langem erwartete Festplatte wird vorerst als 5 1/4-Zoll Laufwerk mit formatiert 20 MByte nur in ganz geringen Stückzahlen ab Anfang März ausgeliefert. Der endgültige Preis stand noch nicht fest. Mittlerweile gibt es wieder eine neue TOS-Version auf Diskette, die mit der künftigen ROM-Version identisch sein soll. Zumindest hat sie eine Länge von knapp 192 KByte. Diese neue Version soll aber weitgehend mit der derzeitigen Version (197744 Bytes) identisch sein, da nur der Text verkürzt wurde. Die ROMs werden in Amerika und England bereits ausgeliefert, bei uns wird es noch ca. vier Wochen dauern. Das TOS im ROM wurde so konzipiert, daß nach jedem Einschalten bzw. Reset zuerst ein Diskettenlaufwerk angesprochen wird, um festzustellen, ob dort ein bootfähiges Betriebssystem vorhanden ist. Ist dies der Fall, so wird es geladen und gestartet. Das hat den Vorteil, daß auch weiterhin andere Betriebssysteme verwendet, sowie Änderungen vorgenommen werden können. Erfreulicherweise soll es zur Hannover-Messe eine ganz neue Basic-Version geben. Diese soll zum jetzigen ST-Basic voll kompatibel sein und zusätzlich noch über einige Befehle mehr verfügen. Ferner soll dieses Basic eine wesentlich größere Genauigkeit und eine höhere Ablaufgeschwindigkeit haben und das, obwohl die Länge des Programms nur noch ca. 80 KByte betragen soll. Nach dem Erscheinen dieses Basics, wird es automatisch beim Kauf eines neuen Rechners mitausgeliefert. Den Besitzern des „alten“ ST-Basics soll der Erwerb gegen eine Selbstkostengebühr ermöglicht werden. Man darf gespannt sein!

Die Frage, ob ein PC-ähnliches Gehäuse für die ST-Rechner geplant sei, wurde uns verneint. Es bestehe aber für die Zukunft die Möglichkeit eines kompatiblen Nachfolgemodells, das den Nachfolgeprozessor 68 020, einen echten 32 Bit-Prozessor, enthalten soll. Dieser neue Rechner wird eine Kapazität von mindestens zwei MByte RAM und eventuell ein UNIX-Betriebssytem erhalten. Ob dieser in einem PC-Gehäuse oder als Zusatzplatine zum jetzigen ST erscheint, steht noch in den Sternen. Im letzteren Fall würde der ST nur noch als Terminal dienen. Das wohlbekannte Betriebssytem MS-DOS (IBM und Kompatible käme wegen Geschwindigkeitsverlusten nur als Hardwarezusatz in Frage. Die Unterstützung in Sachen CP/M bschränkt sich bei ATARI weiterhin nur auf den kostenlosen CP/M-Emulator. Beschaffung weiterer Software bleibt den Anwendern überlassen.

Das von ATARI vertriebene ST Manager: Text wird derzeit überarbeitet, so daß die bekannten Mängel behoben werden, wie zum Beispiel das Anzeigen einer Directory aus dem Programm heraus, was zur Zeit ja noch nicht möglich ist. Diese verbesserte Version soll dann gegen Einsendung des Originals direkt bei der Firma SM in München eingetauscht werden. Nähere Einzelheiten, wie zum Beispiel Ergänzungen zum Handbuch und Liefertermin, waren noch nicht bekannt.

Im großen Ganzen war die Micro ’86 weniger besucht, als im letzten Jahr. Dies lag nicht zuletzt daran, daß die eigentlichen Neuheiten wohl erst auf der Cebit in Hannover vom 12.3. – 19.3. präsentiert werden.

Der ATARI geht fremd

oder: Zwei 3 1/2-Zoll Fremdlaufwerke am ST

Viele Hersteller bieten preiswerte Diskettenlaufwerke an, die untereinander kompatibel sind, so daß ein Anschluß an den ATARI ST durchaus denkbar ist. Wie die Praxis jedoch zeigt, ist die Sache nicht ganz so einfach. Die Probleme, die entstehen, und wie sie zu lösen sind, dies ist in diesem Beitrag sehr genau beschrieben.

In der Januarausgabe wurde bereits die Möglichkeit, ein 5 1/4-Zoll-Laufwerk anzuschließen, erklärt. Interessant ist dies im Zusammenhang mit CP/M oder alten 5 1/4-Zoll-Laufwerken bzw. Disketten und natürlich durch die Tatsache, daß dann IBM-Daten gelesen werden können. In dieser Ausgabe sollen nun **zwei** doppelseitige (1 Megabyte) 3 1/2-Zoll-Laufwerke an den ATARI ST angeschlossen werden. Besonders für die Besitzer eines 260 ST ergibt sich so eine sehr preiswerte Möglichkeit, zwei absolut zuverlässige, lautlose und zudem in den Abmessungen sehr kleine Laufwerke an den Rechner anzuschließen.

Wo liegen die Schwierigkeiten?

Die Verwendung von Fremdlaufwerken, ohne ein ATARI SF 354 bzw. SF 314, bringt vor allem zwei Probleme mit sich. Erstens muß man sich den ungewöhnlichen, 14-poligen DIN-Floppystecker, selber „bauen", zweitens ist das Signal zur Untercheidung de Diskettenseite SIDE SELECT (SS) im Rechner nicht „gepuffert", so daß dieses Signal beim Anschluß zweier doppelseitiger Fremdlaufdwerke überlastet ist. Vollständigkeitshalber muß erwähnt werden, daß sogar drei Leitungen des ST-Floppybusses nicht mit den notwendigen Treiberbausteinen versehen sind. Es sind die Signale SS, DRIVE SELECT 0 (DS0) und DRIVE SELECT 1 (DS1), die übrigens alle vom Soundchip (!) des Rechners erzeugt werden. Die beiden zuletzt genannten Signale DS0 bzw. DS1, die zur Auswahl eines der Laufwerke dienen, benötigen keinen Treiber, da sie

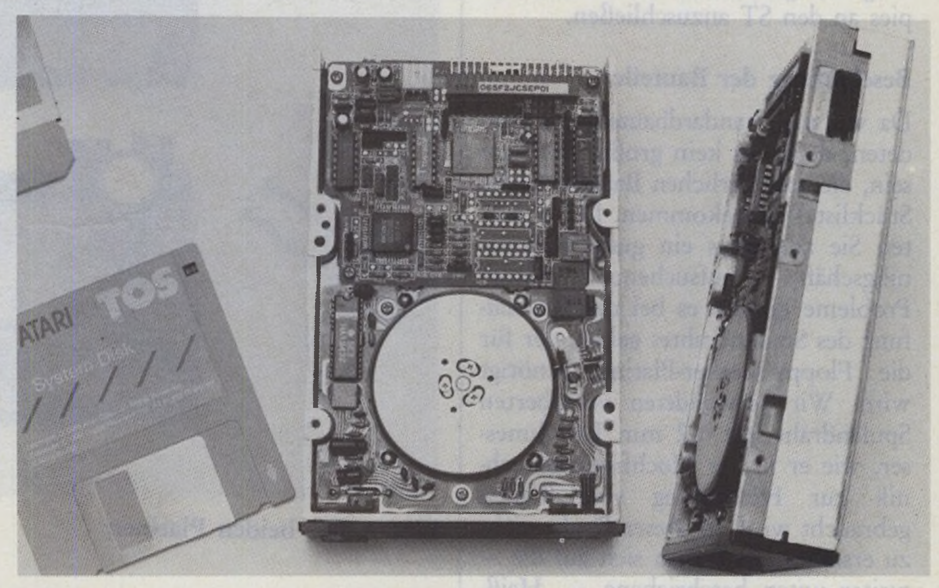


Bild 1: Zwei 3 1/2-Zoll CHINON-Laufwerke

immer nur an ein Laufwerk angeschlossen werden (Unterschiedliches Setzen eines Jumpers auf den Floppies). Aufmerksame Leser werden sich nun fragen: warum funktioniert es dann mit den ATARI Floppies SF 314? Dies liegt an der Tatsache, daß in der Eingangsstufe des ATARI-Laufwerkes „hochohmige" Pull-up-Widerstände (2,7k bis 5,6k Ohm) vorhanden sind. Ein Pull-up-Widerstand ist ein Widerstand, der ein Signal (eine Leitung) auf einen definierten Zustand bringt (auf high Potential d. h. +5 Volt). Bei den meisten 3 1/2-Zoll-Fremdfloppies werden hingegen 1k Ohm Pull-up-Widerstände verwendet, das ergibt bei zwei Laufwerken (Parallelschaltung) einen Widerstand von nur 500 Ohm. Dieser Widerstandswert ist zu klein für den Rechner; die Folge ist, daß das Side-Select-Signal nicht mehr auf low Potential (0 Volt) „heruntergezogen" werden kann.

Was tun?

Wenn es möglich ist, sollten Sie einfach den Pull-up-Widerstand der Side-Select-Leitung aus einer Ihrer 3 1/2-Zoll-Floppies entfernen oder den Widerstandswert erhöhen. Dies ist jedoch leider nur sehr schwer möglich, da bei den 3 1/2-Zoll-Floppies die Pull-up-Widerstände meist nicht mehr steckbar sind, wie z. B. bei unserer 5 1/4-Zoll-Floppy der Januarausgabe. Die sicherste Lösung unseres Problems it die: wir müssen dem Side-Select (SS)-Signal einen Treiber (Verstärker) spendieren. Zu diesem Zweck haben wir eine kleine Schaltung entwickelt und eine Platine dazu entworfen (siehe Bild), die unser Problem in eleganter Weise löst. Auf der Platine ist nicht nur der Treiber für die SS-Leitung untergebracht, sondern auch noch das Netzteil, das die benötigten Spannungen (+5 und +12 Volt) für zwei Laufwerke liefert. Auf der Platine sind ferner eine Leuchtdiode als Einschaltkontrolle sowie zwei Pfostenstecker, die in Form und Anschlußbelegung mit denen an den Floppies übereinstimmen, vorhanden. Der eine Pfostenstecker wird nun über ein 34-poliges Flachbandkabel mit den zwei Laufwerken verbunden (siehe Bild 2), der andere kann ebenfalls über ein Flachbandkabel mit dem Rechner verbunden werden. Hier begegnen wir dem zweiten Problem, nämlich dieser fremden DIN-Buchse. Aber auch hierfür haben wir eine Platine entworfen (siehe Bild 2), die zwei Fliegen mit einer Klappe schlägt. Einerseits bildet sie

einen 14-poligen DIN-Stecker, andererseits bietet diese Platine die Möglichkeit eines 34-poligen Standard-Floppybusses. Mit diesen beiden Platinen und einem 34-poligen Flachbandkabel mit zwei Quetschbuchsen ist es also möglich, gefahrlos zwei Fremdfloppies an den ST anzuschließen.

Beschaffung der Bauteile

Da wir nur Standardbauteile verwendeten, dürfte es kein großes Problem sein, die erforderlichen Bauteile (siehe Stückliste) zu bekommen. Dabei sollten Sie allerdings ein gutes Elektronikgeschäft aufsuchen. Kleine Probleme könnte es bei der Beschaffung des Spulendrahtes geben, der für die Floppy-Stecker-Platine benötigt wird. Wir verwendeten versilberten Spulendraht mit 1,2 mm Durchmesser, wie er in der Hochfrequenztechnik zur Herstellung von Spulen gebraucht wird. Ist dieser Draht nicht zu erstehen, muß man sich auf die – weiter unten beschriebene – „Heißklebertechnik" beschränken.

Die Platinen

Der etwas versiertere Elektronik-Bastler kann sich die Kupferplatinen selbst herstellen, da das Original-Layout im Maßstab 1 : 1 abgedruckt ist. Derjenige, der diese Möglichkeit nicht besitzt, kann die fertigen Platinen auch beim Heim-Verlag bestellen. Der genaue Preis lag bei Redaktionsschluß leider noch nicht vor (etwa DM 20,–).

Die Schaltung

Um die Abmessungen der Platine und den Preis klein zu halten, besteht die Schaltung nur aus einem Treiber IC und einem einfachen Netzteil. Aus diesem Grund werden auch die beiden Ausgangsspannungen von +5 und +12 Volt nur aus einer Wechselspannungsquelle erzeugt. Dies bedeutet aber, daß der 5 Volt-Regler (IC1) eine relativ hohe Verlustleistung zu „verbraten" hat, er muß deshalb gut gekühlt werden. Optimal wäre ein Trafo, der eine Wechselspannung von 13 - 14 Volt liefern würde. Da ein solcher aber nur schwer zu bekommen sein dürfte, kann bei einer guten Kühlung von IC1 auch ein Trafo mit einer Sekundärspannung von 15 Volt verwendet werden. Ist der benötigte Ausgangsstrom der 12-Volt-Spannung nicht zu groß, reicht (wie bei unserem

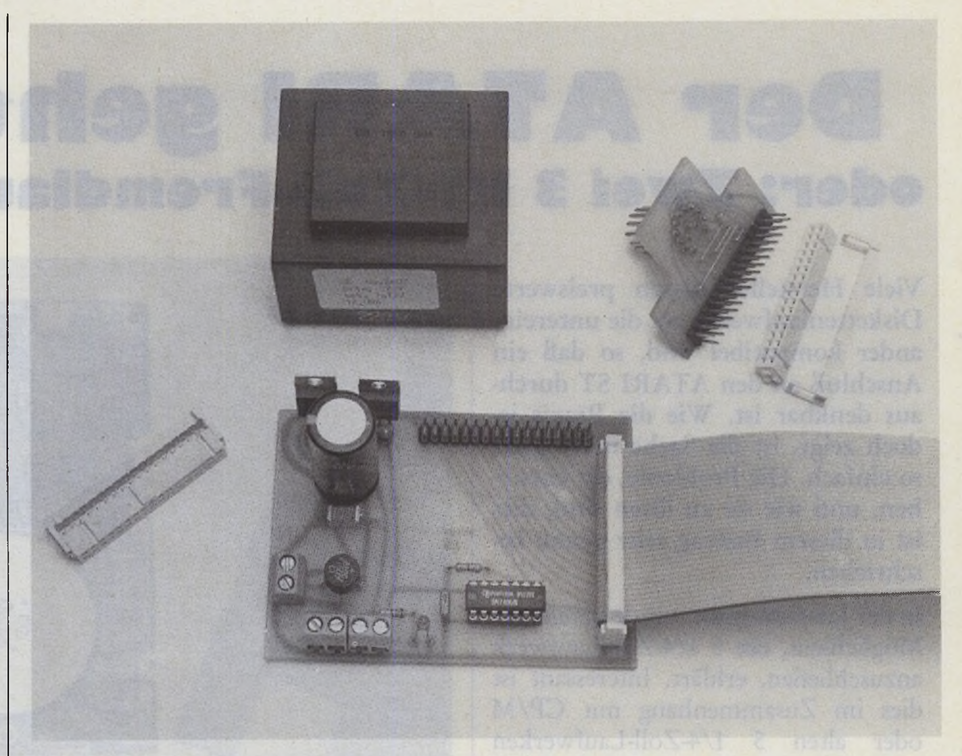


Bild 2: Die beiden Platinen

Stückliste für Netzteil und Treiber:

Bauteil	Anzahl	Wert / Typ	Nr.
Widerstände:	1	1,5 k Ohm	R1
	1	3,3 k Ohm	R2
Kondensatoren:	1	47 nF	C5
	1	220 nF	C1
	2	10 uF/16V (Tantal)	C3,C4
	1	2 200 uF/25V (Elko, stehend)	C2
Halbleiter:	1	Brückengleichrichter B40 C1500 (rund)	GL
	1	IC SN 7407 (Treiber, Open Collector)	IC3
	1	IC Spannungsstabilisator 5 Volt (7805)	IC1
	1	IC Spannungsstabilisator 12 Volt (7812)	IC2
	1	Leuchtdiode (grün, 3 mm Durchmesser)	D1
Sonstiges:	1	doppelreihiges Pfostenleiste (min. 21 polig)	FL
	1	einreihige Pfostenleiste (min. 13 polig)	ST
	6	Lüsterklemmen für Printmontage	LK
	1	Netztrafo 12-14 V / ca. 1 Ampere (= 12 VA)	TR
	1	Netzleitung	
	1	doppelter Ein-Aus Schalter (220 V)	S
	1	IC Fassung Dil 14	
	1	Platine (ST 002 ub)	
	3	Quetschbuchsen 34-polig	
	30 cm	34-poliges Flachbandkabel	
	evtl. 2	Kühlkörper für Spannungsstab.	
	evtl. 1	Gehäuse (für 2 Floppies und Netzteil)	

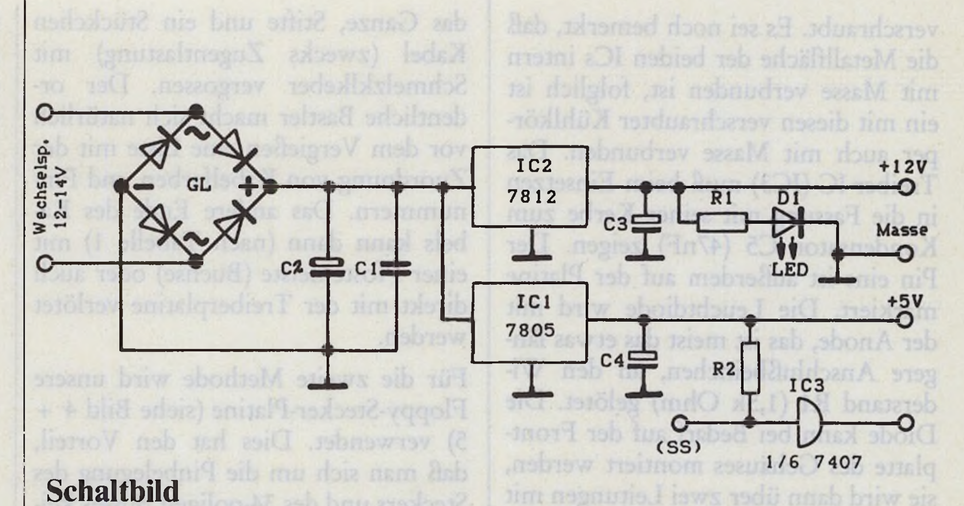
Floppy-Stecker-Platine:

Bauteil	Anzahl	Wert / Typ
Sonstiges:	1	doppelreihige Pfostenleiste (min. 17 polig)
	ca. 30 cm	versilberten Spulendraht 1,2 mm ∅
	1	Platine (ST 001 ub)
	2	Quetschbuchsen 34-polig
	1 m	34-poliges Flachbandkabel

Labormuster) ein Trafo mit 12 Volt Ausgangsspannung aus. Bei diesem Muster konnten wir mit einer Aluplatte von ca. 10 x 10 cm als Kühlfläche bei +5 V einen Strom von ca. 350 mA und bei +12 V einen von ca. 400 mA entnehmen. Das reicht normalerweise für zwei 3 1/2-Zoll Floppies aus. Bei den größeren 5 1/4-Zoll Laufwerken ist die Stromabgabe nur noch für eine Floppy ausreichend. Wie auf den Fotos zu erkennen ist, sind bei unserem Labormuster alle Verbindungen steckbar ausgeführt. Da diese Quetschverbindungen nicht gerade billig sind, kann der geübte Bastler eventuell die eine oder andere Verbindung weglassen und ein Kabel direkt an die Platine löten.

Aufbau und Bestücken der Treiberplatine

Anhand des Schaltbildes, der Stückliste und der Bilder dürfte das Bestücken der Platine keine Schwierigkeiten bereiten. Man muß allerdings auf die richtige Polung der drei Elkos achten; da jedoch auf der Platine jeweils der Pluspol (+) markiert ist, wie auch bei den Elkos, sollte keine Verwechslung der Anschlüsse möglich sein. Der Gleichrichter (GL) mit seinen vier Anschlüssen ist so einzulöten, daß der mit (+) gekennzeichnete Anschluß auch mit dem entsprechend markierten Loch der Platine übereinstimmt. Die beiden ICs zur Spannungsstabilisation IC1 und IC2 zeigen mit ihrer Kühlfläche (Metallfläche) nach außen. Während des Betriebs müssen diese durch eine ausreichend große Metallfläche (ca. 15² cm) gekühlt werden. Dies gilt besonders für das 5-Volt-Stabilisator-IC. Dazu werden die ICs mit jeweils einer kleinen Schraube (M3) mit der Gehäuserückwand oder einem Kühlkörper



Schaltbild

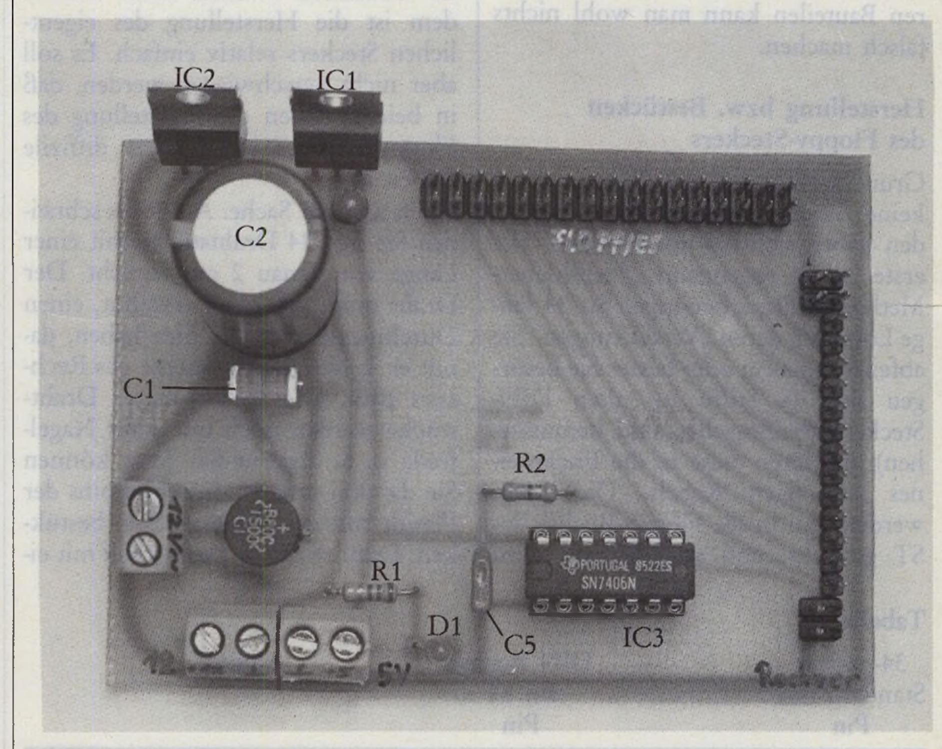
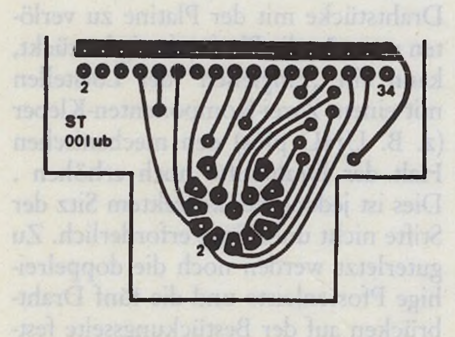
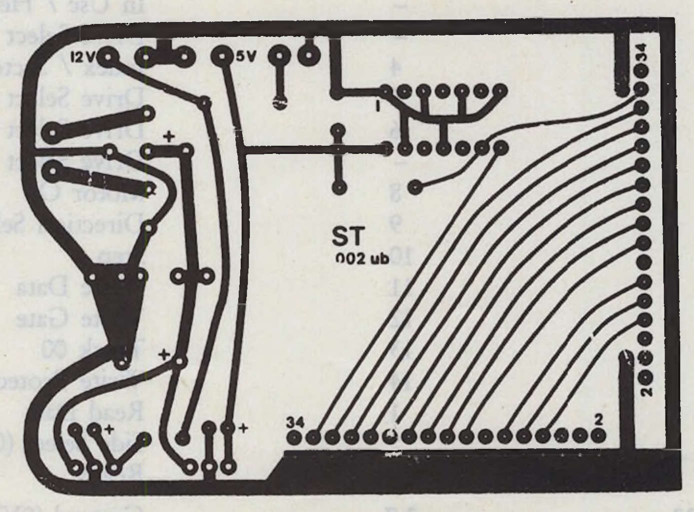


Bild 3: Bestärkungsseite



Layout der FSP



Layout der Treiber- und Netzteilplatine

verschraubt. Es sei noch bemerkt, daß die Metallfläche der beiden ICs intern mit Masse verbunden ist, folglich ist ein mit diesen verschraubter Kühlkörper auch mit Masse verbunden. Das Treiber-IC (IC3) muß beim Einsetzen in die Fassung mit seiner Kerbe zum Kondensator C5 (47nF) zeigen. Der Pin eins ist außerdem auf der Platine markiert. Die Leuchtdiode wird mit der Anode, das ist meist das etwas längere Anschlußbeinchen, an den Widerstand R1 (1,5k Ohm) gelötet. Die Diode kann bei Bedarf auf der Frontplatte des Gehäuses montiert werden, sie wird dann über zwei Leitungen mit der Platine verbunden. Bei allen anderen Bauteilen kann man wohl nichts falsch machen.

Herstellung bzw. Bestücken des Floppy-Steckers

Grundsätzlich sind der Phantasie hier keine Grenzen gesetzt. Zwei Methoden haben sich bei uns bewährt. Die erste ist die sogenannte „Heißkleber-Methode". Dazu benötigen Sie 14 lange Lötnägel, deren „Verdickungen" Sie abfeilen müssen; oder besser Sie besorgen sich 14 Stifte aus alten DIN-Steckern (heißmachen und herausziehen) und löten diese an die Enden eines 14-poligen Kabels. Die Stifte werden nun in die Floppy-Buchse des ST gesteckt, und anschließend wird das Ganze, Stifte und ein Stückchen Kabel (zwecks Zugentlastung) mit Schmelzklkeber vergossen. Der ordentliche Bastler macht sich natürlich vor dem Vergießen eine Liste mit der Zuordnung von Kabelfarben und Pinnummern. Das andere Ende des Kabels kann dann (nach Tabelle 1) mit einer Pfostenleiste (Buchse) oder auch direkt mit der Treiberplatine verlötet werden.

Für die zweite Methode wird unsere Floppy-Stecker-Platine (siehe Bild 4 + 5) verwendet. Dies hat den Vorteil, daß man sich um die Pinbelegung des Steckers und des 34-poligen Busses keine Gedanken machen muß. Außerdem ist die Herstellung des eigentlichen Steckers relativ einfach. Es soll aber nicht verschwiegen werden, daß in beiden Fällen die Herstellung des Floppy-Steckers eine etwas diffizile Arbeit ist.

Doch nun zur Sache. Als erstes schneiden Sie sich 14 Drahtstücke mit einer Länge von genau 2 cm zurecht. Der Draht muß, wie oben erwähnt, einen Durchmesser von 1,2 mm haben, damit er in die Floppy-Buchse des Rechners paßt. Die Enden dieser Drahtstücke werden noch mit einer Nagelfreile o. ä. abgerundet. Jetzt können Sie die drei inneren Anschlußpins der Platine mit den Drahtstücken bestükken. Dazu wird ein Drahtstück mit einer Flachzange von der Lötseite her so durch ein Loch der Platine gesteckt, daß der Draht knapp 1 mm auf der Bestückungsseite herausragt (siehe Bild 4). In gleicher Weise ist mit allen anderen Pins zu verfahren. Es ist darauf zu achten, daß die Löcher der Platine nicht zu groß gebohrt werden (1,2 mm), da sonst ein guter mechanischer Sitz der Drahtstücke nicht gewährleistet ist, und ein bloßer Halt durch Verlöten könnte die Kupferbahnen von der Epoxyd-Platine abreißen. Aus diesem Grund ist es auch sinnvoll, erst vier Drahtstücke einzustecken, auszurichten und dann erst die ersten drei Drahtstücke mit der Platine zu verlöten usw.. Ist die Platine fertig bestückt, kann ein „Eingießen" der Lötstellen mit einem Zwei-Komponenten-Kleber (z. B. UHU plus) den mechanischen Halt der Drahtstifte noch erhöhen . Dies ist jedoch bei korrektem Sitz der Stifte nicht unbedingt erforderlich. Zu guterletzt werden noch die doppelreihige Pfostenleiste und die fünf Drahtbrücken auf der Bestückungsseite festgelötet.

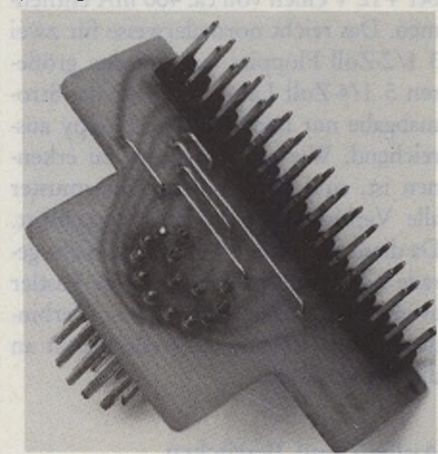


Bild 4: Bestückungsseite (FSP

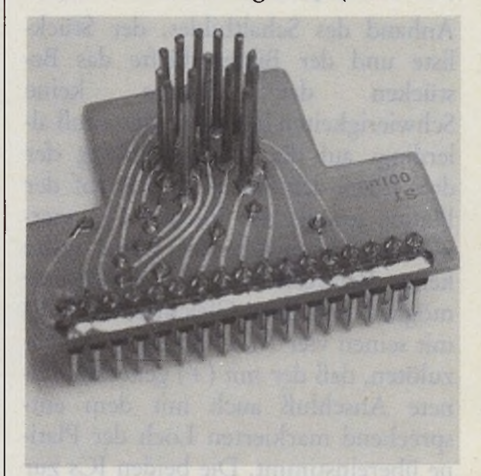


Bild 5: Lötseite (FSP)

Tabelle 1

34-poliger Standard Bus Pin	DIN Buchse am ST Pin	Signal
2	–	–
4	–	In Use / Head Load
6	–	Drive Select 3
8	4	Index / Sector
10	5	Drive Select 0
12	6	Drive Select 1
14	–	Drive Select 2
16	8	Motor On
18	9	Direction Select
20	10	Step
22	11	Write Data
24	12	Write Gate
26	13	Track 00
28	14	Write Protect
30	1	Read Data
32	2	Side Select (0)
34	–	Ready
1,3–,31,33	3,7	Ground (0V)

Der Zusammenbau aller Komponenten

Der fertige Floppystecker und die Treiberkarte werden über ein 34-poliges Flachbandkabel, das an beiden Enden mit einer Quetschbuchse versehen ist, derart miteinander verbunden, daß eine 1 : 1 Verbindung besteht, d. h. Pin 1 der Floppy-Platine wird wieder mit Pin 1 der Treiberkarte verbunden usw.. Ebenso ist die Verbindung von Treiberkarte zu den beiden Floppies zu gestalten. Hierbei ist kaum etwas falsch zu machen. Etwas mehr Sorgfalt ist hingegen bei der Stromversorgung angebracht. Ein Vertauschen der +5 und +12 Volt Spannung würde den sicheren Tod Ihrer Floppy-Laufwerke bedeuten. Vergewissern Sie sich deshalb ganz genau, wo bei Ihren Floppies die +5 bzw. die +12 Volt anzuklemmen sind. Die Treiberkarte benötigt eine Wechselspannung von 12 bis 14 Volt und einen Strom von max. 1 Ampere. Diese Spannung wird von der Sekundärseite eines Trafos, der in das Gehäuse der Floppies eingebaut werden kann, abgenommen. An die Primärseite des Trafos wird über einen Schalter, der auf der Rückseite des Gehäuses montiert wird, ein Netzkabel angelötet. Doch bitte beachten Sie, daß unbedingt alle Leitungen, die eine Spannung von 220 Volt (!) führen, gut zu isolieren sind. Selbstverständlich darf nur an den Leitungen „gearbeitet" werden, solange der Netzstecker gezogen ist!

Bevor man einen ersten Probelauf startet, ist eine Floppy auf Drive Select 0 (DS0) und die andere auf Drive Select 1 (DS1) einzustellen. Dies ist durch Umstecken eines Jumpers (kleiner Verbindungsstecker) auf den Floppies möglich. Nach Kontrolle aller Verbindungen sollte zuerst das Netzteil geprüft werden. Zu diesem Zweck trennt man die Verbindung zu den Floppies ab und misst nach dem Einschalten die Spannungen an den Lüsterklemmen. Liegen die richtigen Spannungswerte vor (+5 und +12 Volt), so kann nun ein Laufwerk angeschlossen werden. Dieses sollte nach dem Einschalten der Spannungsversorgung und des Rechners das TOS booten. Funktioniert jetzt noch alles einwandfrei, dann kann eine leere Diskette formatiert werden. Erscheint kurz darauf die Meldung, daß ca. 720 KBytes freier Speicher auf der Diskette zur Verfügung stehen, so hat man die erste Hürde erfolgreich überwunden. Erscheint jedoch eine Meldung, die einen wesentlich größeren Speicherraum zu Verfügung stellt, so ist das meist ein Zeichen dafür, daß das Side-Select-Signal die Floppy nicht oder nicht korrekt erreicht. In diesem Fall muß der Treiberbaustein überprüft werden. Ist bis jetzt alles positiv verlaufen, so kann nun, nach dem Ausschalten aller Geräte, auch die zweite Floppy angeschlossen und in gleicher Weise, wie gerade beschrieben, verfahren werden. Wahrscheinlich dürfte jetzt nichts mehr schiefgehen, und Sie sind stolzer Besitzer zweier 1-Megabyte-Laufwerke. Bei uns arbeiten seit geraumer Zeit zwei CHINON-Laufwerke des Typs F-354C einwandfrei. Selbstverständlich funktionieren auch andere Laufwerke wie z. B. TEAC FD 35 F mit der hier beschriebenen Schaltung am ATARI ST. Denkbar ist auch der Anschluß zweier 5 1/4-Zoll-Floppies, wie das FD-55 FV von TEAC (Januarausgabe). Eine Bezugsquelle für die oben genannten Laufwerke ist folgende Adresse:

Peter Gebhart
– Microsysteme –
Lange Straße 59
6370 Oberursel

(UB)

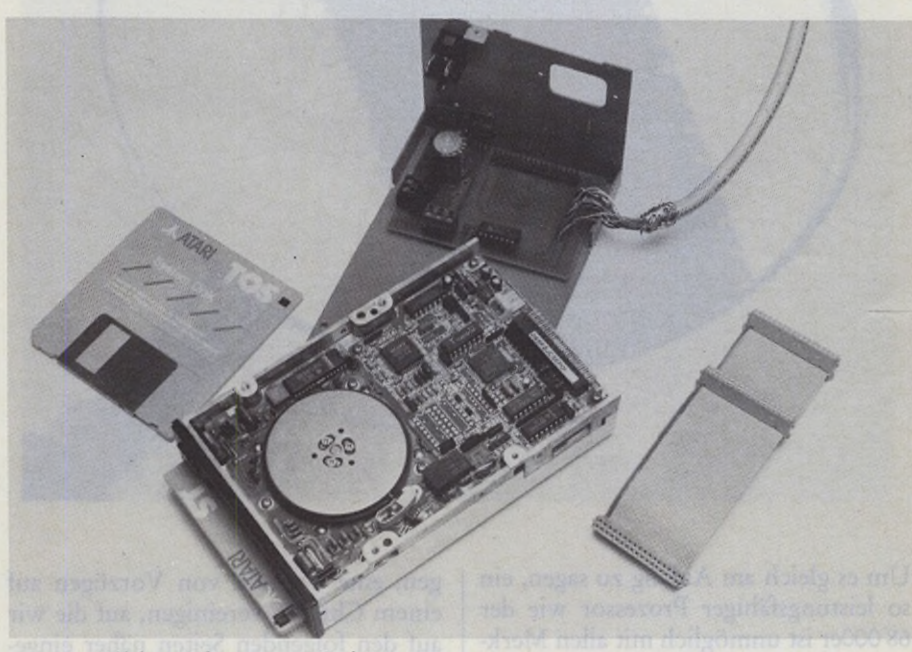


Bild 6: Floppy + Gehäuse

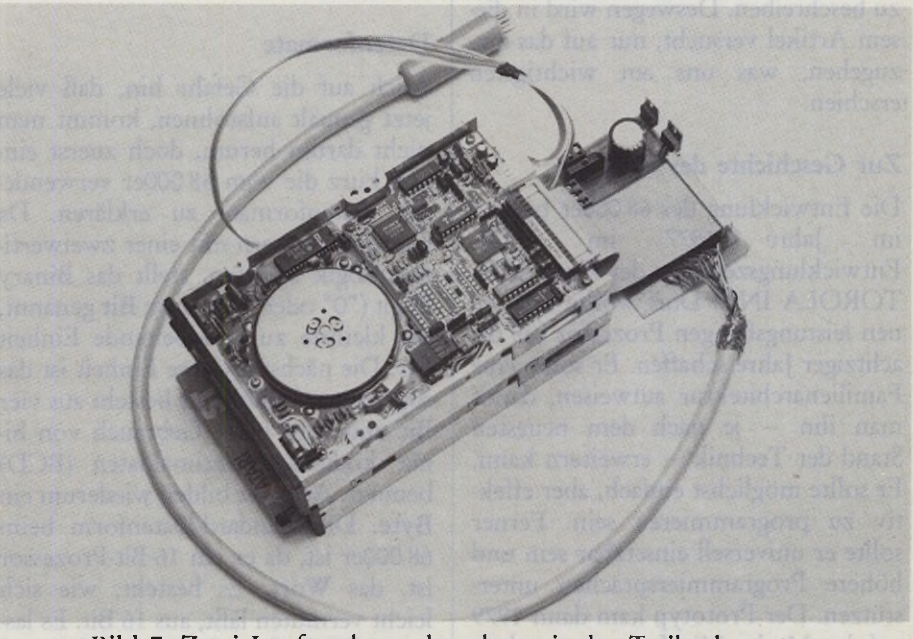


Bild 7: Zwei Laufwerke verbunden mit der Treiberkarte

Den 68000er unter die Lupe genommen



Um es gleich am Anfang zu sagen, ein so leistungsfähiger Prozessor wie der 68 000er ist unmöglich mit allen Merkmalen und Details auf ein paar Seiten zu beschreiben. Deswegen wird in diesem Artikel versucht, nur auf das einzugehen, was uns am wichtigsten erschien.

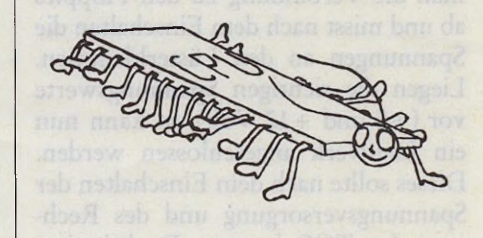
Zur Geschichte des 68 000er

Die Entwicklung des 68 000er begann im Jahre 1977 im MOS-Entwicklungszentrum der Firma MOTOROLA INC. Dort wollte man einen leistungsfähigen Prozessor für die achtziger Jahre schaffen. Er sollte eine Familienarchitektur aufweisen, damit man ihn – je nach dem neuesten Stand der Technik – erweitern kann. Er sollte möglichst einfach, aber effektiv zu programmieren sein. Ferner sollte er universell einsetzbar sein und höhere Programmiersprachen unterstützen. Der Prototyp kam dann 1979 auf den Markt. Mit ihm war es gelungen, eine Vielzahl von Vorzügen auf einem Chip zu vereinigen, auf die wir auf den folgenden Seiten näher eingehen wollen.

Datenformate

Auch auf die Gefahr hin, daß viele jetzt gequält aufstöhnen, kommt man nicht darum herum, doch zuerst einmal kurz die vom 68 000er verwendeten Datenformate zu erklären. Da Computer intern mit einer zweiwertigen Logik arbeiten, stellt das Binary Digit ("0" oder "1"), kurz **Bit** genannt, die kleinste zu verarbeitende Einheit dar. Die nächst größere Einheit ist das sogenannte **Nibble**. Es besteht aus vier Bit und wird beim Gebrauch von binär kodierten Dezimaldaten (BCD) benutzt. Acht Bit bilden wiederum ein **Byte**. Die Standard-Datenform beim 68 000er ist, da er ein 16-Bit-Prozessor ist, das **Wort**. Es besteht, wie sich leicht vermuten läßt, aus 16 Bit. Es lassen sich jedoch auch 32 Bit auf einmal verarbeiten. In diesem Falle spricht man von einem **Langwort**.

Es sei noch gesagt, daß Zugriffe auf Worte und Langworte nur von geraden Adressen erfolgen dürfen. Wird diese Bedingung nicht erfüllt, wird eine sogenannte Ausnahmebehandlung (Exception) ausgeführt.



Die Anschlußbelegung

Der Prozessor hat 64 Anschlüsse, die man in neun Gruppen einteilen kann. (Eine Abbildung der Anschlußbelegung kann man im nebenstehenden Fenster sehen.) Die erste Gruppe besteht aus den 16 Datenleitungen (DO – D15), auf denen die Daten byte- oder wortweise übertragen werden.

Die zweite Gruppe besteht aus den 23 Adreßleitungen (A1 – A 23). Auf ihnen werden die Adressen, die von dem Prozessor erzeugt wurden und zur Auswahl einer der Speicherstellen dienen, transportiert. Diese Adressen legen Quelle und Ziel der über den Datenbus laufenden Daten fest. Mit den 23 Adreßleitungen wäre sogar eine direkte Adressierung von bis zu 16 MByte möglich. Daran ist allerdings beim ST nicht zu denken, da die übrige Hardware dies nicht zuläßt.

Über die Adresse A0 ist zu sagen, daß sie nicht nach außen geführt ist, im Chip selbst aber natürlich existiert. Sie ist hinter den beiden Signalen **U**pper **D**ate **S**trobe (Oberes Daten-Strobe, UDS) und **L**ower **D**ate **S**trobe (Unteres Daten-Strobe, LDS) verborgen, womit wir bei der nächsten Gruppe wären. Diese dienen mit den Signalen Adreß-Strobe (AS), **D**a**T**a **ACK**nowledge (Datenquittierung, DTACK) und den Read/Write (Lese-/Schreibsignal, R/W) der asynchronen Bussteuerung, durch die der Speicher und die 16-Bit-Peripherie angeschlossen werden.

Das Signal Adreß-Strobe zeigt an, daß sich eine gültige Adresse auf dem Adreßbus befindet. Erkennt der 68000er während eines Lesezykluses Data-Acknowledge, werden die betreffenden Daten, die sich auf dem Datenbus befinden, übernommen. Durch das Read/Write-Signal wird bestimmt, ob die Datenübertragung ein Lese- oder Schreibvorgang ist. Ferner wird, zusammen mit beiden Daten-Strobes festgelegt, ob die obere (Bits 0–7) oder die untere (Bits 8–15) angesprochen werden (siehe Tabelle 1).

Gruppe vier beinhaltet die Signale, die mit der Busverwaltung zu tun haben. Es sind **B**us **R**equest (Busanforderung, BR), **B**us **G**rant (Buszuteilung, BG) und **B**us **G**rant **A**cknowledge (Buszuteilungsquittierung, BGACK).

Pin	Signal	Pin	Signal
1	D4	64	D5
2	D3	63	D6
3	D2	62	D7
4	D1	61	D8
5	D0	60	D9
6	$\overline{AS}$	59	D10
7	$\overline{UDS}$	58	D11
8	$\overline{LDA}$	57	D12
9	R/$\overline{W}$	56	D13
10	$\overline{DTACK}$	55	D14
11	$\overline{BG}$	54	D15
12	$\overline{BGACK}$	53	GND
13	$\overline{BR}$	52	A23
14	VCC	51	A22
15	CLK	50	A21
16	GND	49	VCC
17	$\overline{HALT}$	48	A20
18	$\overline{RESET}$	47	A19
19	$\overline{VMA}$	46	A18
20	E	45	A17
21	$\overline{VPA}$	44	A16
22	$\overline{BERR}$	43	A15
23	$\overline{IPL2}$	42	A14
24	$\overline{IPL1}$	41	A13
25	$\overline{IPL0}$	40	A12
26	FC2	39	A11
27	FC1	38	A10
28	FCO	37	A9
29	A1	36	A8
30	A2	35	A7
31	A3	34	A6
32	A4	33	A5

Anschlußbelegung des 68000

Durch Bus Request wird dem Prozessor mitgeteilt, daß eine Peripherie-Einheit den Bus benutzen will. Bus Grant zeigt allen anderen Busbenutzern an, daß der Prozessor nach Beendigung des derzeitigen Zykluses die Bussteuerung abgibt. Durch Bus Grant Acknowledge wird erkannt, daß eine andere Einheit die Bussteuerung übernommen hat. Dazu müssen aber zuvor die Signale AS, DTACK und BGACK inaktiv und BG aktiv werden.

Die drei **I**nterrupt **P**riority **L**ine-Signale (IPLO – LPL2) bilden die fünfte Gruppe. Durch sie werden die einzelnen Interrupt-Ebenen (zusammen sieben Möglichkeiten) bestimmt, auf die wir noch näher bei der Exceptionbesprechung eingehen wollen. Es sei nur gesagt, daß dem Anwender insgesamt 199 Interruptvektoren zur Verfügung stehen.

Die nächste Gruppe bilden die sogenannten Function Code Signale (FC0 – FC1). Diese zeigen in kodierter Form den momentanen Status und die des gerade ausgeführten Zykluses an (siehe Tabelle 2). Die Signale sind immer dann gültig, wenn das Adreß-Strobe-Signal aktiv ist. Zu dem Unterschied zwischen User- und Supervisor-Modus kommen wir noch bei der Besprechung der Register.

Die siebte Gruppe mit den Signalen VPA, E und VMA dienen zur synchronen Bussteuerung. Das **V**alid **P**eriphal **A**dress-Signal (Gültige Peripherieadresse, VPA) zeigt an, daß die Datenübertragung mit dem Freigabe-Signal Enable (E) synchronisiert werden muß. Außerdem gibt es an, daß das betreffende Peripheriegerät über einen sanchronen Bus verfügt. Das **V**alid **M**emory **A**dress-Signal (Gültige Speicheradresse) meldet, daß der Prozessor synchronisiert ist und sich eine gültige Adresse auf dem Adreßbus befindet.

Die vorletzte Gruppe bilden die Systemsteuerungs-Signale. Sie unterschieden sich in BERR, RESET und HALT. Das **B**us **ERR**or-Signal (Busfehler, BERR) kann durch mehrere Fehlerarten ausgelöst werden. Einmal kann eine Peripherieeinheit nicht antworten oder es kann zum Beispiel eine falsche Interrupt-Prozedur erfolgt sein. Wird BERR LOW, so wird eine Ausnahmebehandlung (Exception) eingeleitet. Falls auch noch das **HALT**-Signal LOW wird, versucht der Prozessor den Befehl, der den Busfehler ausgelöst hat, zu wiederholen. Klappt auch dieser Versuch nicht, tritt ein sogenannter doppelter Busfehler

UDS	LDS	R	W	D8 – D15	D0 – D7
1	1	–	–	keine gült. Daten	keine gült. Daten
0	0	1	0	Datenbits 8 – 15	Datenbits 0 – 7
1	0	1	0	keine gült. Daten	Datenbits 0 – 7
0	1	1	0	Datenbits 8 – 15	keine gült. Daten
0	0	0	1	Datenbits 8 – 15	Datenbits 0 – 7
1	0	0	1	Datenbits 8 – 15	Datenbits 0 – 7
0	1	0	1	Datenbits 8 – 15	Datenbits 0 – 7

Tabelle 1: Datenbussteuerung mit den Signalen UDS, LDS und R/W.

auf und der Prozessor wird angehalten.

Man sollte eigentlich meinen, daß der Begriff **RESET** jedem Computer-Anwender geläufig ist. Doch beim 68000er gibt es noch einiges dazu zu sagen. Durch einen externen RESET (zum Beispiel mittels des RESET-Knopfes) werden der Prozessor und die ganze Peripherie in ihren Einschaltzustand versetzt. Nun ist bei diesem Prozessor auch noch ein durch Software erzeugter, interner RESET möglich. Durch ihn werden lediglich die externen Peripherie-Einheiten zurückgesetzt. Um dies zu erreichen, gibt es den Assemblerbefehl RESET.

Die letzte Gruppe von Anschlüssen ist an jedem Prozessor vorhanden. E ist das **CLocK**-Signal (Takt, CLK), die Spannungsversorgung **Vcc** und die Massezuleitung **GND**. Die Taktfrequenz des Prozessors beträgt beim ATARI ST 8 lMHz.

FC0	FC1	FC2	Status
0	0	0	undefiniert
0	0	1	Daten im User-Modus
0	1	0	Programm im User-Modus
0	1	1	undefiniert
1	0	0	undefiniert
1	0	1	Daten im Supervisor-Modus
1	1	0	Programm im Supervisor-Modus
1	1	1	Interruptquittierung

Tabelle 2: Statusmeldungen durch die Signale FC0 – FC 2.

Ausnahmebehandlungen (Exceptions)

Exceptions sind durch aufgetretene Fehler oder interne Befehle ausgelöste Ausnahmebehandlungen. Zu ihnen zählen auch die Interrupts und Traps, auf die wir gleich noch kommen. Jeder Exception ist ein bestimmter Exception-Vektor mit eigener Vektornummer zugeordnet. Für diese Vektoren sind die Adressen $000 – $3FC im Arbeitsspeicher vorgesehen (siehe Tabelle). Beim Eintreten einer Exception holt sich der Prozessor automatisch den Inhalt der Adresse, die im betreffenden Vektor steht, in den Programmzähler. Dazu sei noch erwähnt, daß so die "Pilze" bzw. "Bomben" beim Absturz des Systems entstehen. Die Anzahl dieser Symbole gibt nämlich die Vektornummer des Fehlers an, der diese Ausnahmebehandlung ausgelöst hat.

Die einzelnen Vektoren haben folgende Bedeutung:

Der **RESET (SSP)**-Vektor beinhaltet die Adresse, deren Inhalt nach einem Hardware-RESET in den Supervisor-Stackpointer geladen wird.

Im **RESET (PC)**-Vektor steht die Adresse, deren Inhalt nach einem Hardware-RESET in den Programmzähler geladen wird.

Auf den **Busfehler** sind wir ja schon vorher kurz eingegangen. Er wird immer von einem externen Gerät ausgelöst.

Ein **Adreßfehler** tritt dann auf, wenn bei einer Wort- oder Langwortverarbeitung der Zugriff auf eine ungerade Adresse stattfindet.

Belegung der Ausnahmevektoren beim ATARI ST

Vektornummer	Adresse hex	Zuordnung
0	000	Anfangsadresse SSP nach Reset
1	004	Anfangsadresse PC nach Reset
2	008	Busfehler
3	00C	Adreßfehler
4	010	unzulässiger Befehl
5	014	Division durch Null
6	018	Befehl CHK
7	01C	Befehl TRAPV
8	020	Privilegverletzung
9	024	Trace
10	028	1010 Emulator (Line A)
11	02C	1111 Emulator (Line F)
12	030	reserviert
13	034	reserviert
14	038	reserviert
15	03C	nicht initialisierter Int.-Vektor
16 - 23	040 - 05C	reserviert
24	060	Unechter Interrupt
25	064	Interrupt-Autovektor (Ebene 1)
26	068	Interrupt-Autovektor (Ebene 2)
27	06C	Interrupt-Autovektor (Ebene 3)
28	070	Interrupt-Autovektor (Ebene 4)
29	074	Interrupt-Autovektor (Ebene 5)
30	078	Interrupt-Autovektor (Ebene 6)
31	07C	Interrupt-Autovektor (Ebene 7)
32	080	TRAP 0
33	084	TRAP 1 (GEMDOS)
34	088	TRAP 2 (GEM)
35	08C	TRAP 3
36	090	TRAP 4
37	094	TRAP 5
38	098	TRAP 6
39	09C	TRAP 7
40	0A0	TRAP 8
41	0A4	TRAP 9
42	0A8	TRAP 10
43	0AC	TRAP 11
44	0B0	TRAP 12
45	0B4	TRAP 13 (BIOS)
46	0B8	TRAP 14 (XBIOS)
47	0BC	TRAP 15
48 - 63	0C0 - 0FC	reserviert
64 - 255	100 - 3FC	Non-Interrupt-Autovektoren

Unter einem **unzulässigen Befehl** versteht man eine Bitkombination, die nicht vom Prozessor entschlüsselt werden kann. Dies kann man zum Beispiel zum Setzen von Breakpoints nutzen. Der Assemblerbefehl ILLEGAL hat einen unzulässigen Befehl zur Folge.

Division durch Null erklärt sich an und für sich von selbst. Wenn bei einer Division im Nenner eine Null steht, ist dies ein mathematischer Fehler und hat eine Exceptionabarbeitung zur Folge.

Durch den Befehl **CHK** werden die Register auf ihre Grenzen geprüft. Befindet sich der Prozessor außerhalb des definierten Bereichs, so wird dieser CHK-Vektor benutzt.

Das **TRAPV**-Vektor tritt dann in Kraft, wenn das Overflow-Flag im Statusregister gesetzt ist und der Befehl TRAPV gerade ausgeführt wird.

Eine **Privilegverletzung** kommt dann vor, wenn ein Anwender einen privilegierten Befehl (zum Beispiel MOVE to SR) ablaufen läßt, sich aber noch im User-Modus befindet.

Ist das **Trace-Flag** im Statusregister gesetzt, wird nach jedem Befehl in das mit dem Trace-Vektor bestimmte Programm gesprungen. So ist eine Einzelschritt-Bearbeitung eines Programms relativ einfach zu bewerkstelligen.

Bei den beiden **Emulator**-Vektoren handelt es sich um die Line A- und Line F-Vektoren. Wird ein nicht vorhandener Befehl abgearbeitet und die höchsten vier Bit enthalten die Kombinationen 1010 (hex. A) oder 1111 (hex. F), so werden diese beiden Vektoren angesprungen.

Der **nicht initialisierte Interrupt**-Vektor dient zum Erkennen unerwünschter Interrupts oder nicht initialisierten Interrupt-Vektor-Registern von Peripheriebausteinen.

Falls ein Busfehler während der Interruptquittierung auftritt, wird der **unechte Interrupt**-Vektor benutzt.

Nun zu den **Interrupt-Autovektoren.** Wie oben schon erwähnt, besitzt der 68 000er sieben Interruptebenen, wobei die siebte die höchste Interruptebene ist. Diese Ebene ist der sogenannte „nicht maskierbare Interrupt" (NMI).

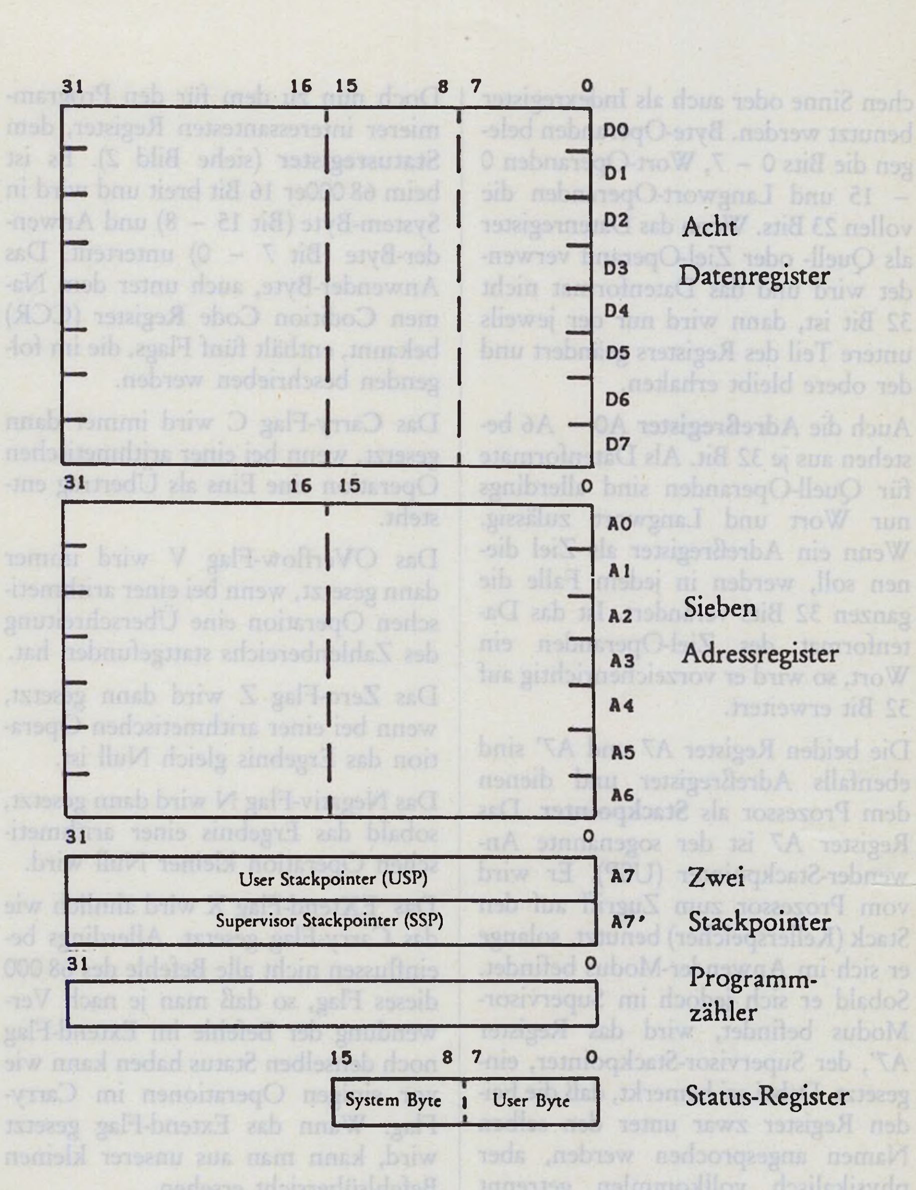


Bild 1: Die Register des 68000

Durch Setzen bzw. Löschen der drei Interruptbits im Statusregister werden die einzelnen Ebenen ausgewählt. Bei den Autovektoren handelt es sich um Vektoren, bei denen der Prozessor sich aus der beteffenden Interruptebene die richtige Vektornummer ableitet. Als Gegenstück dazu liegen am Ende der Vektorenliste 6Adresse $00 – $3FC) die **Non-Autovektoren**, die vom Anwender erzeugt und über den Datenbus übergeben werden müssen.

Dem ST stehen allerdings nur die Ebenen 2, 4 und 6 zur Verfügung. Diese Vektoren zeigen auf die Horizontal/Vertikal-Blank-Routinen und die vom MFP-Baustein verwalteten Interrupts.

Als letztes werden die **TRAP**-Vektoren besprochen. Sobald der Prozessor einen TRAP-Befehl abarbeitet, lädt er den betreffenden TRAP-Vektor in den Programmzähler und startet das an dieser Adresse stehende Programm. Beim ST werden allerdings nur vier von den 15 TRAP-Vektoren benutzt. So ruft TRAP #1 das GEMDOS, TRAP #2 das GEM, TRAP #13 das BIOS und TRAP #14 das XBIOS auf.

Die Register

Der 68 000er verfügt über acht 32 Bit-Datenregister, sieben 32 Bit-Adreßregister, zwei 23 Bit-Stackpointer, einen 32 Bit-Programmzähler und ein 16 Bit-Statusregister (siehe Bild 1).

Die acht **Datenregister** DO – D7 sind vielseitig verwendbar. Sie können entweder als Datenregister im übli-

chen Sinne oder auch als Indexregister benutzt werden. Byte-Operanden belegen die Bits 0 – 7, Wort-Operanden 0 – 15 und Langwort-Operanden die vollen 23 Bits. Wenn das Datenregister als Quell- oder Ziel-Operand verwendet wird und das Datenformat nicht 32 Bit ist, dann wird nur der jeweils untere Teil des Registers geändert und der obere bleibt erhalten.

Auch die **Adreßregister** A0 – A6 bestehen aus je 32 Bit. Als Datenformate für Quell-Operanden sind allerdings nur Wort und Langwort zulässig. Wenn ein Adreßregister als Ziel dienen soll, werden in jedem Falle die ganzen 32 Bits verändert. Ist das Datenformat des Ziel-Operanden ein Wort, so wird er vorzeichenrichtig auf 32 Bit erweitert.

Die beiden Register A7 und A7' sind ebenfalls Adreßregister und dienen dem Prozessor als **Stackpointer**. Das Register A7 ist der sogenannte Anwender-Stackpointer (USP). Er wird vom Prozessor zum Zugriff auf den Stack (Kellerspeicher) benutzt, solange er sich im Anwender-Modus befindet. Sobald er sich jedoch im Supervisor-Modus befindet, wird das Register A7', der Supervisor-Stackpointer, eingesetzt. Dabei sei bemerkt, daß die beiden Register zwar unter den selben Namen angesprochen werden, aber physikalisch vollkommlen getrennt sind.

Der **Programmzähler** (PC) sei mehr oder wenig nur zur Ergänzung erwähnt. Er beinhaltet immer die nächste abzuarbeitende Adresse des laufenden Programms.

Doch nun zu dem für den Programmierer interessantesten Register, dem **Statusregister** (siehe Bild 2). Es ist beim 68 000er 16 Bit breit und wird in System-Byte (Bit 15 – 8) und Anwender-Byte (Bit 7 – 0) unterteilt. Das Anwender-Byte, auch unter dem Namen Codition Code Register (CCR) bekannt, enthält fünf Flags, die im folgenden beschrieben werden.

Das Carry-Flag C wird immer dann gesetzt, wenn bei einer arithmetischen Operation eine Eins als Übertrag entsteht.

Das OVerflow-Flag V wird immer dann gesetzt, wenn bei einer arithmetischen Operation eine Überschreitung des Zahlenbereichs stattgefunden hat.

Das Zero-Flag Z wird dann gesetzt, wenn bei einer arithmetischen Operation das Ergebnis gleich Null ist.

Das Negativ-Flag N wird dann gesetzt, sobald das Ergebnis einer arithmetischen Operation kleiner Null wird.

Das EXtend-Flag X wird ähnlich wie das Carry-Flag gesetzt. Allerdings beeinflussen nicht alle Befehle des 68 000 dieses Flag, so daß man je nach Verwendung der Befehle im Extend-Flag noch denselben Status haben kann wie vor einigen Operationen im Carry-Flag. Wann das Extend-Flag gesetzt wird, kann man aus unserer kleinen Befehlsübersicht ersehen.

Die übrigen Bits 5 – 7 sind nicht benutzt.

Im System-Byte werden ebenfalls nur fünf Bits verwendet. Da wären zunächst einmal die Bits 8 – 10. Bei ihnen handelt es sich um die drei Interrupt-Flags. Sie werden, je nachdem, welche Interruptebene man wünscht, gesetzt. Wie oben bereits gesagt, gibt es sieben Interruptebenen, was sich daraus ergibt, daß man mit drei Bit maximal eine Sieben (binär = 111) darstellen kann.

Die anderen beiden Bits zeigen an, ob der Supervisor- (Bit 13) und der Trace-Modus (Bit 15) eingeschaltet sind. Auf den Trace-Modus sind wir ja schon oben eingegangen. Doch der Begriff Supervisor-Modus steht bis jetzt noch ungeklärt im Raum. Um es einfach auszudrücken: der Unterschied zwischen dem Supervisor- und dem Anwender-Modus besteht darin, daß im Supervisor-Modus alle Befehle benutzt werden dürfen, im Anwender-Modus dagegen alle, bis auf die priviligierten Befehle, die in unserer Liste mit einem „Priv" versehen wurden. Versucht man trotzdem, diese im Anwender-Modus zu benutzen, wird automatisch eine Ausnahmebehandlung (Privilegverletzung) ausgelöst. Die Unterscheidung zwischen den beiden Modi wurde deswegen gemacht, weil in einem eventuellen Mehrbenutzer-System nicht jeder Anwender auf alle Teile des Betriebssystems freien Zugriff haben soll. Nach einem RESET befindet sich der Rechner immer im Supervisor-Modus.

Die Adressierungsarten

Bevor wir zu den Adressierungsarten kommen, müssen wir noch kurz klären, was **effektive Adressen** (EA) sind. Bei den meisten Befehlen wird

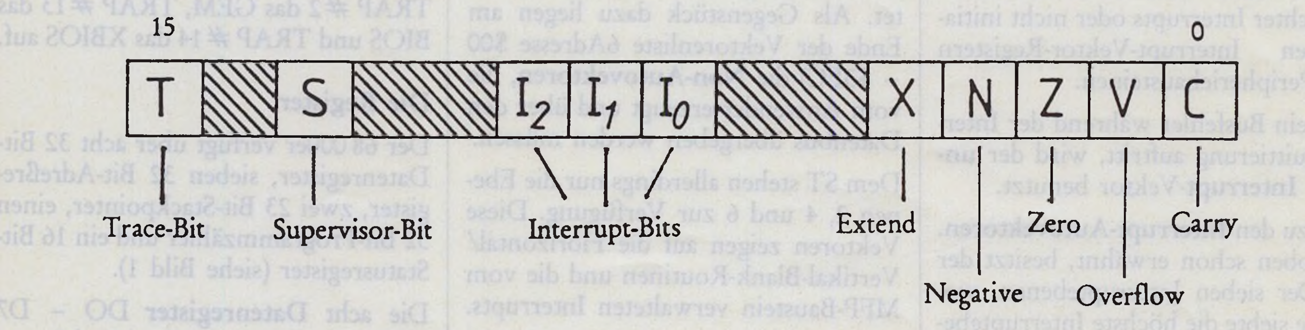


Bild 2: Das Statusregister des 68000

die Adresse des Operanden durch das effektive Adreßfeld im Operationscode bestimmt. In diesem Adreßfeld ist die effektive Adresse enthalten. Sie besteht aus zwei Hälften, dem Modusfeld und dem Registerfeld, wobei das Modusfeld die Adressierungsart und das Registerfeld die Nummer des gewählten Registers bestimmt.

Der 68 000er bietet die Möglichkeit, 13 verschiedene Adressierungsarten zu verwenden. Diese lassen sich dann wieder in fünf Hauptgruppen einteilen.

Gruppe 1: **Register direkt**

Wenn diese Adressierungsart gewählt wird, muß der Operand in einem der 16 Daten- und Adreßregister abgelegt sein.

Datenregister direkt DN
Adreßregister direkt An

Gruppe 2: **Register indirekt**

Bei dieser Adressierungsart ist der Operand im Speicher abgelegt. Seine Adresse ist in dem Register enthalten, das im Registerfeld der EA steht.

Adreßregister indirekt
(An)

Adreßregister indirekt
mit Postinkrement (An)+

Adreßregister indirekt
mit Preinkrement –(An)

Adreßregister indirekt
mit Adreßdistanz d16(An)

Adreßregister indirekt
mit Index und Adreßdistanz d8(An,Rx)

Gruppe 3: **Daten absolut**

Das Registerfeld in der EA wird in diesem Falle zur Bestimmung der speziellen Adressierungsart verwendet.

Absolute kurze Adresse $XXXX
Absolute lange Adresse $XXXXXXXX

Gruppe 4: **Relativ zu Programmzähler**

Programmzähler
mit Adreßdistanz d16(PC)

Programmzähler
mit Index und Adreßdistanz d8(PC,Rx)

Gruppe 5: **Daten unmittelbar**

Bei dieser Adressierungsart sind die Daten im nächsten Wort bzw. den beiden nächsten Worten oder bereits direkt im Operationscode untergebracht.

Unmittelbar
#X, #XX, #XXXX oder #XXXXXXXX

Unmittelbar schnell
#X, #XX, #XXXX oder #XXXXXXXX

Kurze Assembler-Befehlsliste

In der Liste werden folgende Abkürzungen verwendet:

Operationen

An = Adreßregister n
C = Carry-Flag
CC = Carry-Flag gelöscht
CCR = Anwender-Byte des Statusregisters
Dn = Datenregister n
ea = effektive Adresse
Priv = privilegierter Befehl
Q = Quelloperand
Rx = Register x
Ry = Register y
SP = Stackpointer
SR = Statusregister
SSP = Supervisor-Stackpointer
USP = Anwender-Stackpointer
V = Overflow-Flag
X = Extend-Flag
Z = Zieloperand

Flaggen

★ = Flag wird beeinflußt
– = Flag wird nicht beeinflußt
0 = Flag wird gelöscht
U = Flag ist unbestimmbar

Mnemonic	Beschreibung	Operation	Flaggen X N Z V C
1. Additions-Befehle:			
ABCD	Addiere dezimal mit Carry	(Z)+(Q)+X = Z	* U * U *
ADD	Addiere binär	(Z)+(Q) = Z	* * * * *
ADDA	Addiere Adresse	(Z)+(Q) = Z	- - - - -
ADDI	Addiere unmittelbar	(Z)+Daten = Z	* * * * *
ADDQ	Addiere schnell	(Z)+Daten = Z	* * * * *
ADDX	Addiere mit Carry	(Z)+(Q) = Z	* * * * *
2. Subtraktions-Befehle:			
SBCD	Subtrahiere dezimal mit Carry	(Z)-(Q)-X = Z	* U * U *
SUB	Subtrahiere binär	(Z)-(Q) = Z	* * * * *
SUBA	Subtrahiere Adresse	(Z)-(Q) = Z	- - - - -
SUBI	Subtrahiere unmittelbar	(Z)-Daten = Z	* * * * *
SUBQ	Subtrahiere schnell	(Z)-Daten = Z	* * * * *
SUBX	Subtrahiere mit Carry	(Z)-(Q)-X = Z	* * * * *
3. Multiplikations-Befehle:			
MULS	Multiplikation mit Vorzeichen	(Z)*(Q) = Z	- * * 0 0
MULU	Multiplikation ohne Vorzeichen	(Z)*(Q) = Z	- * * 0 0
4. Divisions-Befehle:			
DIVS	Division mit Vorzeichen	(Z)/(Q) = Z	- * * * 0
DIVU	Division ohne Vorzeichen	(Z)/(Q) = Z	- * * * 0
5. Negations-, Clear- und Erweiterungs-Befehle:			
NBCD	Negiere dezimal mit Carry	0-(Z)DEZ-X = Z	* U * U *
NEG	Negiere	0-(Z) = Z	* * * * *
NEGX	Negiere mit Carry	0-(Z)-X = Z	* * * * *
CLR	Lösche einen Operanden	0->Z	- 0 1 0 0
EXT	Vorzeichenerweiterung	(Z)vorzeichener-weitert -->Z	- * * 0 0
6. Logische-Befehle:			
AND	Logisches UND	(Z)&(Q) = Z	- * * 0 0
ANDI	Logisches UND unmittelbar	(Z)& Daten = Z	- * * 0 0
EOR	Logisches EXOR	(Z)#(Q) = Z	- * * 0 0
EORI	Logisches EXOR unmittelbar	(Z)# Daten = Z	- * * 0 0
NOT	Einer-Komplement	(Z)` = Z	- * * 0 0
OR	Logisches ODER	(Z)$(Q) = Z	- * * 0 0
ORI	Logisches ODER unmittelbar	(Z)$ Daten = Z	- * * 0 0
7.Shift-und Rotate-Befehle:			
ASL	Arithmetisches Linksverschieben	X,C<-(-Z-)<-0	* * * * *

Mnemonic	Beschreibung	Operation	Flaggen X N Z V C
ASR	Arithmetisches Rechtsver-schieben	->(-Z-)->C,X <---	* * * * *
LSL	Logisches Linksver-schieben	X,C<-(-Z-)<-0	* * * 0 *
LSR	Logisches Rechtsver-schieben	0->(-Z-)->C,X	* * * 0 *
ROL	Rundverschieben Links ohne Carry	C<-(-Z-)<- -------->	- * * 0 *
ROR	Rundverschieben Rechts ohne Carry	->(-Z-)->C <-------	- * * 0 *
ROXL	Rundverschieben Links mit Carry	<-X,C<-(-Z-)<- -------------->	* * * 0 *
ROXR	Rundverschieben Rechts mit Carry	->(-Z-)->C,X-> <------------	* * * 0 *
8. Compare- und Test-Befehle:			
CHK	Prüfe Register auf Grenzen	Wenn Dn<0 oder Dn>(ea) => TRAP	- * U U U
CMP	Vergleiche	(Z)-(Q)	- * * * *
CMPA	Vergleiche Adresse	(Z)-(Q)	- * * * *
CMPI	Vergleiche unmittelbar	(Z)- Daten	- * * * *
CMPM	Vergleiche Speicher	(Z)-(Q)	- * * * *
TAS	Prüfe und setze Operanden	(Z) testen => CC 1->7 des Ziels	- * * 0 0
TST	Prüfe einen Operanden	(Z) testen ->CC	- * * 0 0
9. Bit-Befehle:			
BCHG	Prüfe ein Bit und ändere es		- - * - -
BCLR	Prüfe ein Bit und lösche es		- - * - -

Mnemonic	Beschreibung	Operation	Flaggen X N Z V C
BSET	Prüfe ein Bit und setze es		- - * - -
BTST	Prüfe ein Bit		- - * - -
10. Datentransport-Befehle:			
EXG	Vertausche Daten zwischen Registern	Rx <--> Ry	- - - - -
LEA	Lade die effektive Adresse	Z->An	- - - - -
LINK	Anbinden und reservieren	An->SP-,SP->An SP+d->SP	- - - - -
MOVE	Transportiere Daten von der Quelle zum Ziel	(Q)->Z	- * * 0 0
MOVE to CCR	Transportiere zum Bedingungscode-Register	(Q)->CCR	* * * * *
MOVE to SR	Transportiere zum Status-Register	(Q)->SR	* * * * * Priv
MOVE from SR	Transportiere vom Status-Register	SR->Z	- - - - -
MOVE USP	Transportiere den Inhalt des Anwender-Stackpointers	USP->An An->USP	- - - - - Priv
MOVEA	Transportiere Adresse	(Q)->Z	- - - - -
MOVEQ	Transportiere schnell	Daten ->Z	- * * 0 0
MOVEM	Transportiere mehrere Register	Register ->Z (Q)-> Register	- - - - -
MOVEP	Transportiere periphere Daten	(Q)->Z	- - - - -
PEA	Ablegen der effektiven Adresse	Z->SP-	- - - - -
SWAP	Vertausche Registerhälften	(R31-16)<->(R15-0)	- * * 0 0
UNLK	Lösen	An->SP,SP+->An	- - - - -
11. Bedingte Sprünge und Operationen:			
Bcc	Verzweige bedingt	Wenn cc, dann PC+d->PC	- - - - -
DBcc	Prüfe Bedingung, dekrementiere und verzweige	Wenn cc=>Dn-1=Dn Dn<>-1=>PC+d->PC	- - - - -
Scc	Setze Byte bedingt	cc => Einsen->Z sonst Nullen->Z	- - - - -
"cc" steht stellvertretend für alle anderen bedingten Sprünge (siehe auch Liste der Verzweigungsbedingungen.)			
12. Unbedingte Sprünge:			
BRA	Verzweige immer	PC+d->PC	- - - - -
BSR	Verzweige zur Subroutine	PC->SP- PC+d->PC	- - - - -
JMP	Springe	Z->PC	- - - - -
JSR	Springe zur Subroutine	PC->SP- Z->PC	- - - - -

Mnemonic	Beschreibung	Operation	Flaggen X N Z V C
13. Rücksprünge:			
RTE	Springe von Ausnahme zurueck	SP+->SR SP+->PC	* * * * * Priv
RTR	Springe zurück und lade Bedingungscodes	SP+->CC SP+->PC	* * * * *
RTS	Springe zurück von der Subroutine	SP+->PC	- - - - -
14. Trap-Befehle:			
ILLEGAL	Unzulässiger Befehl	PC->SSP-,SR->SSP- (Vektor)->PC	- - - - -
TRAP	Trap	PC->SSP-,SR->SSP- (Vektor)->PC	- - - - -
TRAPV	Trap bei Überlauf	Wenn V, dann Trap	- - - - -
15. Weitere Befehle:			
NOP	Keine Operation	-	- - - - -
RESET	Setze externe Einheiten zurück	-	- - - - - Priv
STOP	Lade das Status-Register und stoppe	Daten ->SR, Stop	* * * * * Priv

Liste der Verzweigungsbedingungen

BCC	Carry-Flag = 0
BCS	Carry-Flag = 1
BEQ	gleich, Zero-Flag = 1
BGE	größer oder gleich
BGT	größer
BHI	höher
BLE	kleiner oder gleich
BLS	niedriger oder identisch
BLT	kleiner
BMI	Negativ-Flag = 1
BNE	ungleich, Zero-Flag = 0
BPL	Negativ-Flag = 0
BVC	Overflow-Flag = 0
BVS	Overflow-Flag = 1

Manche mögen's bunt

Ein Bericht über den Thomson Farbmonitor und allgemeine Überlegungen über Datensichtgeräte.

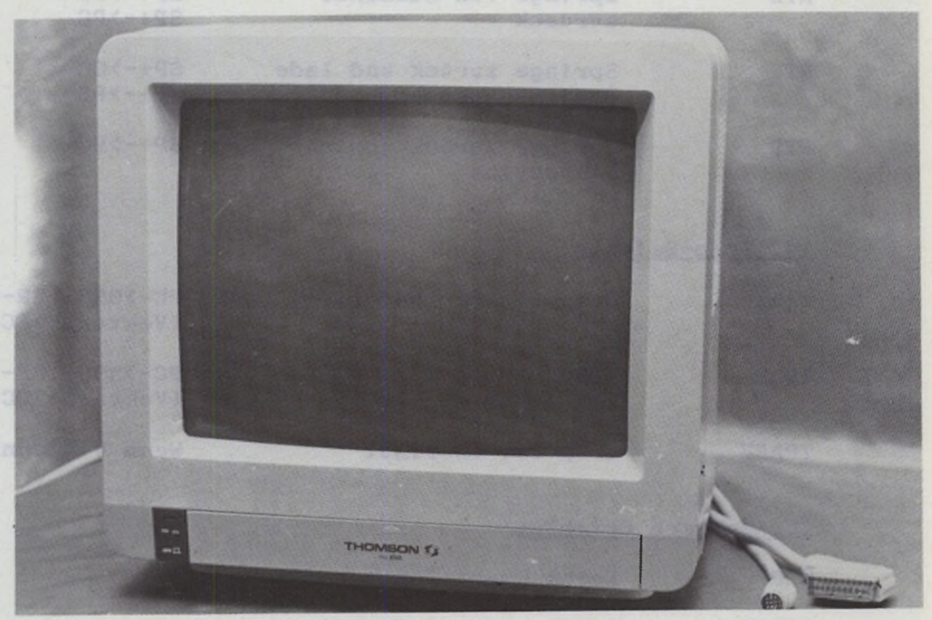


Bild 1: Thomson Farbmonitor

Vor dem Kauf eines Farbmonitors oder einer grundsätzlichen Entscheidung zwischen bunt und schwarzweiß sollte man sich genau darüber im Klaren sein, in welchen Einsatzgebieten der Rechner verwendet wird. Denn wer ausschließlich mit einem Textverarbeitungsprogramm arbeitet, ist mit dem monochromen ATARI Monitor SM 124 ausgezeichnet bedient. Doch meistens sind die Anwendungsgebiete des Rechners nicht so fest fixiert. Allgemein kann man feststellen, daß demjenigen, der vor allem mit Text und Zahlen zu tun hat, der oben erwähnte Schwarzweiß-Monitor mit seiner enormen Schärfe und Auflösung am besten geholfen ist. Liegen Ihre Anwendungen mehr im Bereich der Grafik oder auch in Spielen, so ist ein Farbmonitor durchaus angebracht. Aber, um es gleich vorweg zu sagen, für etwa die gleiche Bildqualität des SM 124 müssen Sie in Farbe schon einige tausend Mark anlegen. Haben Sie sich für einen Farbmonitor entschieden, wäre dies sicher nicht der richtige Moment, um Geld zu sparen, denn die Arbeit vor der „Röhre" sollte ja Spaß machen und nicht zur Quälerei der Augen werden.

Damit Sie die technischen Daten eines Monitors (egal ob bunt oder schwarzweiß) besser verstehen können, wollen wir hier in kurzer Form die wichtigsten Merkmale erläutern. Die Bildröhre ist neben der Elektronik ganz entscheidend für die Bildqualität. Sie „schreibt" mittels eines Elektronenstrahles ein Bild, das Zeile für Zeile von oben nach unten aufgebaut wird. Dies geschieht so schnell, daß das träge menschliche Auge von diessem zeilenweisen Bildaufbau (fast) nichts merkt. Die Leuchtschicht der Bildröhre besteht meistens aus Stoffgemischen mit Phosphor und bestimmt unter anderem auch die Nachleuchtdauer des Bildes. Es ist festzuhalten, daß, je schneller bzw. öfter ein Bild pro Sekunde aufgebaut (geschrieben) wird, es desto flimmerfreier wird. Diese Tatsache spiegelt sich in der **Vertikalfrequenz** oder auch **Bildfrequenz** wieder, je höher sie ist, desto flimmerfdreier ist das Bild. Selbstverständlich spielt hier die Nachleuchtdauer der Leuchtschicht auch noch eine Rolle. Eine lange Nachleuchtdauer verhindert aber schnelle Bildwechsel, da dann das alte Bild noch sichtbar ist. Für einen guten Monitor ist bei normaler Nachleuchtdauer eine Bildfrequenz von 70 Hz notwendig. Leider wird die Bildfrequenz (aus Kostengründen) oft direkt von der Netzfrequenz abgeleitet, die hierzulande 50 Hz beträgt, was dann 50 Bildwechseln pro Sekunde entspricht. Dieser kleine Wert kann besonders im Randbereich zu Flimmereffekten führen.

Neben dem Flimmern ist die Schärfe des Bildes ein wichtiges Kriterium für die Bildqualität. Sie wird durch verschiedene Faktoren beeinflußt.

aussetzung für eine gute Bildschärfe ist ein hoher Kontrast des Bildes. Doch was bedeutet eigentlich ein hoher Kontrast? Kontrast ist die Fähigkeit, nebeneinanderliegende Bildpunkte mit unterscchiedlichen Helligkeitswerten, im Extremfall abwechselnd schwarze und weiße kleine Punkte, noch konturenscharf wiederzugeben, ohne daß dabei die Punkte zu einem einheitlichen grau verschwimmen. Ein Computerbild kann nun, da es digital erzeugt wird, extreme Helligkeitsunterschiede zwischen zwei benachbarten Bildpunkten erzeugen, das Videosignal weist dann eine enorme Flankensteilheit auf (Rechtecksignal). Diese Tatsache unterscheidet ein Computerbild z. B. von dem eines Videorecorders, bei dem zwei Punkte mit großem Helligkeitsunterschied immer ineinander verschwimmen. Um ein solches kontrastreiches Computerbild auch wiedergeben zu können, muß der Monitor eine extrem große **(Video-) Bandbreite** besitzen. Je größer die Bandbreite ist, desto feiner sind die Strukturen, die das Bild wiedergeben kann. Mit den heutigen Bauelementen ist es allerdings relativ einfach, dem Videoverstärker eine große Bandbreite zu verpassen. Für ein gestochen scharfes Bild ist aber nicht nur eine große Bandbreite des Videoverstärkers erforderlich, vielmehr muß auch die restli-

Neue Dimensionen für ATARI-Anwender

Dysans Geschichte ist genauso unglaublich, wie die des Atari ST. Gegründet im Jahre 1973 zur Entwicklung und Produktion von Testgeräten für Speichermedien.

Es folgte die Produktion von bitfehlerfreien Datenstapeln, und dann – im Jahr 1975 – die erste einseitig verwendbare abriebfeste 8" Diskette und die erste Analog-Alignement-Diskette.

Die erste Minifloppy der Welt

1976 brachte Dysan die erste Minifloppy der Welt heraus, eine 5 1/4" Diskette, die inzwischen zum Industriestandard geworden ist.

Im Jahr darauf entwickelte Dysans Forschungs- und Entwicklungsabteilung die erste beidseitig verwendbare 8" Diskette mit doppelter Aufzeichnungsdichte und die erste Minifloppy mit 96/100 Spuren pro Inch.

Die große Neuheit des Jahres 1979 war die erste Digital-Diagnostik-Diskette zur Prüfung der Laufwerk-Justierung und der Leistungsmerkmale.

UHR II High Density-Datenträger

Speziell zur Verwendung mit Laufwerken hoher Dichte, wie sie z. B. der IBM AT aufweist, hat Dysan 1981 die High Density UHR II Diskette entwickelt. Die Richtungsweisende UHR-Technologie von Dysan gestattet Bit-Aufzeichnungsdichten bis zu 18 000 Bit pro Zoll.

DYSAN – ein Name, der für Qualität steht

Dysans guter Ruf gründet auf kompromißlosem Streben nach optimaler Qualität.

Die vollkommen firmeninterne Produktion – angefangen von den selbstentwickelten Beschichtungsformeln bis zum einmaligen Fertigungsprozeß und den sdtrengen Test- und Überwachungsverfahren – garantiert eine nicht zu überbietende Qualitätskontrolle. Hochwertige Produktqualität, innovatives Konzept und weltweit reichhaltigste Auswahl an Hochpräzisions-Speichermedien und verwandten Produkten.

Die Oberflächenstruktur ist ein maßgeblicher Qualitätsfaktor: eine ungleichförmige Oberfläche führt zu erhöhtem Verschleiß des Schreib-/Lesekopfes und abrupte Strukturveränderung bewirken Verlust der Datensicherheit. Zur Gewährleistung einer absolut gleichmäßigen Diskettenoberfläche hat Dysan eine spezielle Polierscheibe mit konischem Profil entwickelt.

Gesamtoberflächen-Test

Bei den als „100 %" bezeichneten Testmethoden vieler Hersteller wird normalerweise nur der Daten-Spurbereich der Diskette geprüft. Die restlichen 40 % der Oberfläche entgehen der Prüfung.

Dysan testet die gesamte Oberfläche, und zwar sowohl die Spuren wie auch die Spurzwischenräume – ein Prüfverfahren, welches beinahe den doppelten Zeitaufwand beansprucht. Wichtig ist diese Gesamtoberflächenprüfung deshalb, weil die Spurenpositionierung des Laufwerkes, das der Kunde verwendet, nicht unbedingt exakt der des Testgerätes entspricht.

Ferner ist es möglich, daß nicht entdeckte Defekte in den Zwischenräumen sich ausdehnen und allmählich auch auf die Datenspuren übergreifen. Bei Dysan heißt 100 % wirklich 100 % geprüft.

Die Dysan Disketten-Serie

Dysan bietet die vollständigste Auswahl an hochqualitativen flexiblen Magnetdatenträgern in der ganzen Welt. Sie umfaßt Standard 8 und 5 1/4" Disketten, Disketten mit 96/100 Spuren pro Zoll und 5 1/4" Disketten mit Ultrahochauflösung (UHR II) sowie 3 1/2" Mikrodisketten.

Alle Datenträger von Dysan werden nach denselben hohen Gütestandards gefertigt. Allen wird nach Bestehen von Dysans einmaligem Gesamtoberflächentest auf missing pulse, extra pulse und Modulation 100 % Fehlerfreiheit bescheinigt, und zwar für die Spuren- und die Zwischenspurbereiche.

Die Dysan 3 1/2" Mikrodiskette

Die Idee bei der Entwicklung von Dysan Mikrodisketten war die Optimierung der Leistungsfähigkeit der führenden Computer auf dem Markt. Die 3 1/2" Dysan Mikrodiskette kann beidseitig bis zu 1 000 000 Datenbytes speichern. Das ist genausoviel wie eine 96 TPI (Spuren pro Zoll) 5 1/4" Diskette.

In Punkto Leistung und Zuverlässigkeit kann sich die Dysan Mikrodiskette durchaus mit den 8 und 5 1/4" Disketten messen. Und, auf Dysan Mikrodisketten sind Ihre Daten in einem festen Plastikumschlag geschützt. Staub und Fingerabdruckproblemen wird mit einem Schiebeverschluß ein Riegel vorgeschoben. Damit ist die Diskette bei Nichtgebrauch automatisch versiegelt. Eine Schreibsperre verhindert versehentliches Löschen Ihrer wertvollen Daten.

ATARI und DYSAN

Dysan Disketten erhalten keine Auszeichnungen, da man hier Anwenderseits den hohen Standard der Güte und Qualität voraussetzt. Anders bei der Hardware. Hier sind die Auszeichnungen ein besonderes Äquivalent für Innovation.

Es sollte deshalb der Anwender, der „ausgezeichnete" Hardware einsetzt, auch ausgezeichnete Software einsetzen.

Deshalb gehört in jeden ATARI Dysan Datenträger.

che Elektronik diesen hohen Anforderungen gewachsen sein, und vor allem muß die Bildröhre diese feinen Strukturen noch auflösen können. Man sollte daher nicht, durch die Angabe einer riesigen Bandbreite, automatisch auf ein sehr scharfes Bild schlußfolgern. Allerdings sind 6 MHz (Megaherz) Bandbreite, wie sie bei normalen Fernsehempfängern zu finden sind, für einen guten Monitor viel zu wenig. Ein solcher sollte mit einer Bandbreite von ca. 15 MHz oder mehr aufwarten können.

Bei einem Farbmonitor besteht nun jeder Bildpunkt aus drei Farbpunkten, die den drei Grundfarben (Rot, Grün und Blau) entsprechen. Je nach Intensität dieser drei Farben entsteht durch Mischung dieser, ein Bildpunkt in einer beliebigen Farbe. Nun können diese drei Farbpunkte nicht beliebig klein bzw. dicht angeordnet werden. Daher ist die **Größe** einer solchen Dreiergruppe, die schließlich einen Bildpunkt ergibt, gleichbedeutend mit der Auflösung bzw. der kleinsten Struktur, die die Bildröhre verarbeiten kann. Es ist daher ein wichtiges Kriterium für die Qualität einer Bildröhre, wie breit eine solche Dreiergruppe ist. Sie wird üblichersweise als Pixelabstand bezeichnet und in Millimeter angegeben (engl. dot pitch). Fernseher haben einen Pixelabstand von ca. 0,65 mm, ein sehr guter Monitor erreicht immerhin 0,28 mm. Oftmals wird die Auflösung auch mit Bildpunkten in horizontaler- und vertikaler Richtung angegeben. Je größer die Zahlenwerte sind, desto besser ist die Auflösung. Sie sollte aber mindestens so gut sein, wie die des Rechners.

Noch ein paar Worte zur Signalübertragung. Es gibt hier verschiedene Normen und Formen, auf die wir hier jedoch nicht alle eingehen wollen. Bekannt ist vor allem das Video-Composite-Signal oder auch BAS- (schwarzweiß) bzw. FBAS-Signal (bunt). Dieses Mischsignal aus Bildinformation, Synchronimpulsen und evtl. Farbinformation ist an die Fernsehnorm gebunden und hat besonders bei Computerbildern den Nachteil, daß die einzelnen Informationen (Impulse) vor der eigentlichen Bildwiedergabe im Monitor wieder getrennt werden müssen. Dies ist besonders bei Farbbildern mit Qualitätsminderung verbunden. Aus diesem Grund wird beim ATARI ST diese Signalform auch nicht verwendet, sondern es werden alle benötigten Signale separat übertragen. Optimal ist die Ansteuerung eines Farbmonitors nur, wenn jede Grundfarbe ihre eigene Leitung besitzt. Man spricht von der sogenannten RGB-Ansteuerung, wie sie beim ST vorhanden ist.

Solche technischen Daten können vor dem Kauf eines Monitors eine gute Richtlinie sein. Es sollte aber auf jeden Fall ein optischer Vergleichstest, auch mit sehr teuren Monitoren gemacht werden.

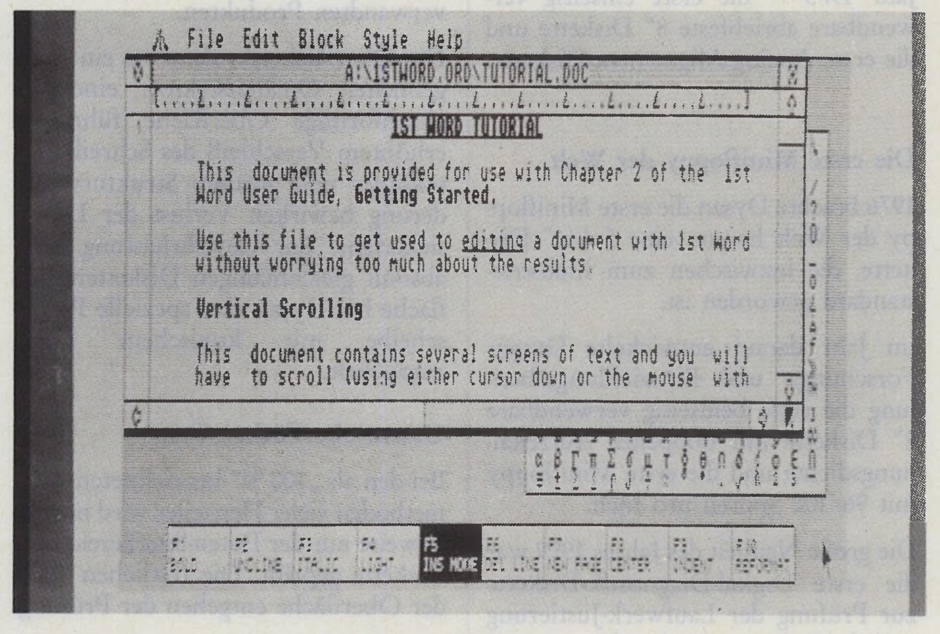


Bild 2: Textbeispiel bei mittlerer Auflösung

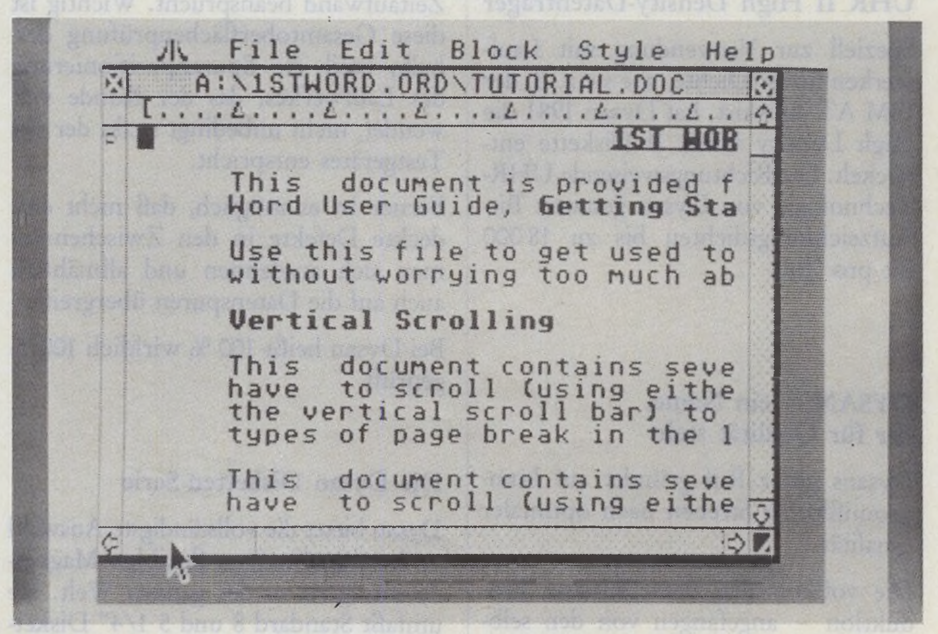


Bild 3: Der gleiche Text bei niedriger Auflösung

Der THOMSON RGB-Farbmonitor CM 36382 AR

Oder auch schlicht „THOMSON for ATARI“ genannt, ist der erste Farbmonitor der von ATARI für die ST-Serie geliefert wird. Der Monitor ist mit der Europa-Normbuchse, der sogenannten SCART-Buchse, ausgestattet. Er wird komplett mit einem zwei Meter langen Anschußkabel geliefert. Dabei fällt positiv auf, daß der Monitorstecker für den ST eine Verriegelung besitzt, so daß er nicht ungewollt herausrutschen kann. Im Betrieb zeigt er sich zuverlässig und bedienerfreundlich, da sämtliche Einstellregler auf der Frontseite untergebracht sind. Allerdings sind nur die wichtigsten Regler, nämlich für Kontrast, Helligkeit und Lautstärke mit „Knöpfen“ bestückt, alle anderen, für vertikale Synchronisation, Bildhöhe und Farbsättigung, können nur unter Zuhilfenahme eines Schraubendrehers verstellt werden. Die Farbwiedergabe ist kräftig und brillant. Die Bildverzerrungen sind auch im Randbereich als sehr gut einzustufen. Allerdings ist ein „Pumpen“ des Bildes bei schnellem Bildwechsel festzustellen, d. h. die Bildgröße schwankt mit der Helligkeit des dargestellten Bildes. Die beiden Bildschirmfotos (leider sind z. Z. noch keine Farbfotos möglich) zeigen die Qualität des Monitors bei niedriger und mittlerer Auflösung. Angenehm ist, daß die Bildröhre entspiegelt ist. Die Standfestigkeit des Monitors könnte etwas besser sein, bei leicht schiefer Standfläche kippt der Monitor nach vorne über. Das Gerät kostet mit Anschlußkabel 1398,– DM. Dieser Preis erscheint uns etwas zu hoch. Zum Schluß noch ein paar technische Daten des Monitors.

Technische Daten:
THOMSON CM 36382 AR

Farbbildröhre:
36 cm diagonal; entspiegelt

Pixelabstand:
0,38 mm

Videobandbreite:
12 MHz (RGB)

Auflösung:
640 x 240 Punkte; mittlere Auflösung

Horizontale- (Zeilen-)Frequenz:
15,625 KHz

Vertikale- (Raster-)Frequenz:
50 Hz

Eingangssignal:
RGB / 75 Ohm

Audio-Ausgang (Ton):
0,85 Watt (max.)

Netzspannung:
220–240 Volt / 50 Hz

Leistungsaufnahme:
54 Watt

Gehäuseabmessung:
37 x 38 x 35 cm (B x T x H)

Gewicht:
9,5 kg



1ST WORD

Version 1.00/1.04

Überblick

1ST WORD ist ein Textverarbeitungsprogramm von GST Holding Ltd. (England). Die erste Version dieses Programms wurde Anfang Februar durch eine neue Version (1.04) ersetzt. Sie wird von ATARI ausgeliefert und ist für DM 99 erhältlich. Zum Schreiben von Serienbriefen läßt sich 1st Word mit dem Programm **1st Mail** kombinieren.

Das Programm wird auf einer Diskette, ohne weitere Dokumentation ausgeliefert. Diese befindet sich als Textfile auf der Diskette.

Mit gewissen Einschränkungen in der graphischen Darstellung läuft 1st Word auch mit einem Farbmonitor in der mittleren und niedrigen Auflösung. Darauf wird im weiteren Text jedoch nicht speziell eingegangen.

Um es vorweg zu sagen, 1ST WORD ist ein wirklich gelungenes Textverarbeitungsprogramm. Es nutzt viele Möglichkeiten des ATARI ST sehr gut aus (siehe Bild 1). Dazu zählen vor allem die graphischen Fähigkeiten (siehe Schriftarten), die gute Menütechnik (siehe Bild 2) und die Möglichkeiten, die sich durch die Verwendung der Maus ergeben.

Da die Anleitung bis jetzt nur in englischer Sprache geliefert wird, ist dieser Bericht auch als Hilfe für den Anwender gedacht, der damit nicht zurechtkommt.

1ST WORD ist sehr leicht zu bedienen, so daß man nach einer kurzen Einarbeitungszeit die Anleitung nur noch selten benutzen wird. Dazu trägt vor allem die gute Menütechnik bei. Sie erleichtert die Benutzung des Programms sehr, weil man sich nicht, wie bei anderen Textverarbeitungsprogrammen, bestimmte Tastenkombinationen merken muß, um bestimmte Funktionen zu erreichen.

Laden von 1st word

Wenn man 1STWORD.PRG im Ordner angewählt hat, wird das Programm und sein GEM Quellfile (.RCS) geladen (wenn man 1STWORD auf eine RAM-Floppy legen will, muß man mindestens diese beiden Teile dorthin laden). Nach dem Laden erscheint auf dem Bildschirm ein Feld mit allen verfügbaren Zeichen und am unteren

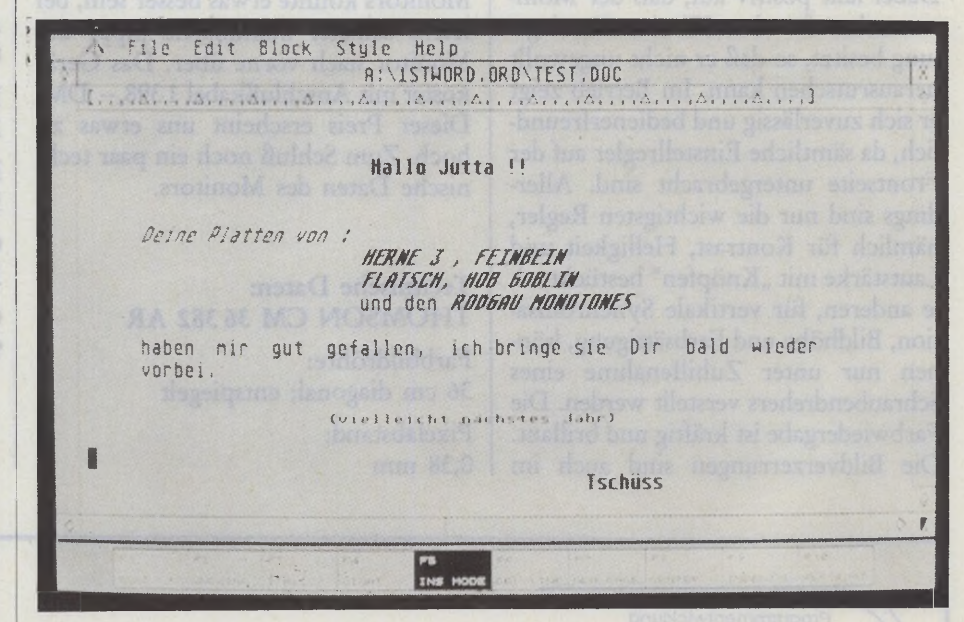


Bild 1

Rand eine Leiste mit den Tastenbelegungen. Außerdem steht in der Mitte ein Lademenue. Dort befinden sich bereits die Dokumente GUIDE.DOC und TUTORIAL.DOC. GUIDE.DOC ist die Anleitung zu 1ST WORD (leider noch in Englisch), TUTORIAL. DOC ist ein Textfile zum Editieren und Üben spezifischer Funktionen des Textverarbeitungsprogramms.

In der Anleitung (GUIDE) kann man

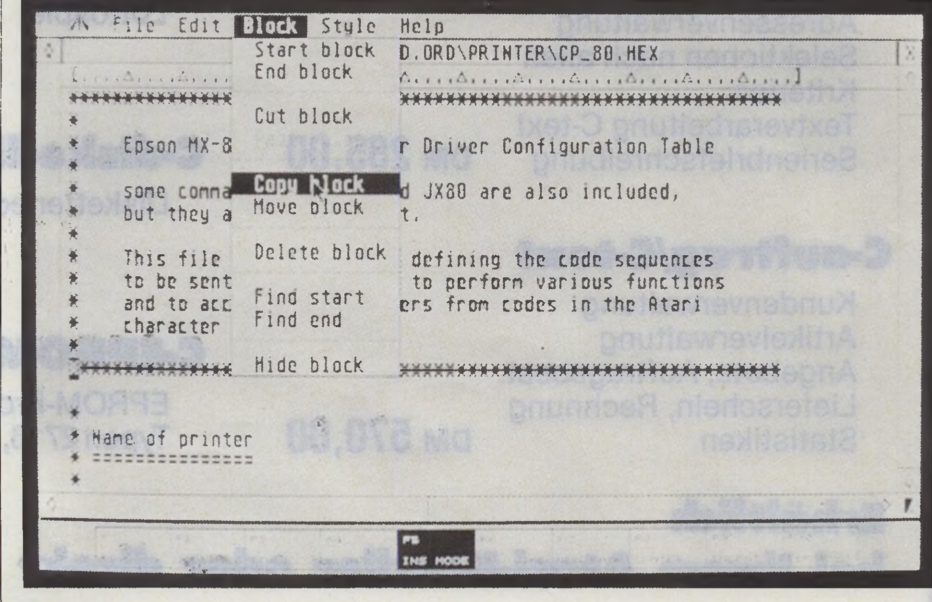


Bild 2

nun nach Belieben herumblättern und sich die wichtigsten Punkte durchlesen. Sinnvoller ist jedoch, sich den ganzen Text ausdrucken zu lassen (etwa 43 Seiten). Wenn man keinen der voreingestellten Drucker besitzt, kann man den Text nur in Normalschrift ausdrucken. Für besondere Schriftarten (Fettdruck, Unterstrichen usw.) muß dann der Drucker angepaßt werden (siehe Druckeranpassung).

Erstellen eines Dokuments (DOC)

Wenn man ein neues Dokument bzw. Textfile erstellen will, gibt man im Lademenü einen neuen Filenamen mit dem Zusatz .DOC ein. Nach anklicken der OK-Taste stellt das Programm fest, daß dieses Textfile noch nicht existiert und fragt, ob es errichtet werden soll. Man antwortet mit OK. Nun erscheint das Textverarbeitungsfenster auf dem Bildschirm. Es hat die von Basic her bekannte Form und kann auch so behandelt werden. Man kann es also mit der Maus verschieben, verkleinern, vergrößern oder ganz löschen (!).

Das Textfenster

Wenn man einen längeren Text geschrieben hat, so kann man ihn mit dem rechten Scroll-Block durchsuchen. Dabei ist ein seitenweises Verschieben und auch das Springen an jede beliebige Stelle des Dokuments möglich, dazu wird der weiße Balken angeklickt und verschoben (Taste gedrückt halten).

Mit dem unteren Block kann man sich dementsprechend Texte ansehen, die mehr als 80 Zeichen pro Zeile haben.

Der Streifen am linken Rand zeigt die Seitennummer an. Hier gibt es drei Arten:

1) Eine Zahl mit durchgezogenem Strich bedeutet, daß hier ein echtes Seitenende ist. Der Drucker würde dann ein neues Blatt anfangen. Diese Seitennummern werden vom Benutzer gesetzt, indem man entweder die Taste F7 drückt oder mit der Maus die Stelle im linken Randstreifen anklickt.
2) Eine Zahl mit einer gestrichelten Linie bedeutet, daß hier das Programm eine neue Seite anfängt, weil die im Punkt **Layout** festgelegte Zeilenzahl überschritten wurde.

Die Hauptmenüs

1) ATARI:	wie bei anderen ATARI-Programmen können sich hier die Desk-Accessoirs befinden
2) FILE:	hier sind die Funktionen zum Laden, Speichern und zum Drucken enthalten
3) EDIT:	hier kann eine der wichtigsten Funktionen eingeschaltet werden, der Wordprocessing Mode (WP); er aktiviert den Menüpunkt STYLE Außerdem befinden sich hier Funktionen zum Suchen, Ersetzen und zum Setzen von Marken
4) BLOCK:	hier sind die Funktionen zum Bewegen, Kopieren und Löschen von Textblöcken
5) STYLE:	hier können die Schriftarten und Funktionen wie Zentrieren, Zeilenausgleich und Wortumbruch gewählt werden
6) HELP:	hier werden bestimmte Funktionen des Textverarbeitungsprogramms erklärt

Das FILE-Menü

OPEN	– ruft das Lademenü auf
PRINT	– ruft das Druckermenü auf (siehe unten)
SAVE	– speichert das Textfile unter seinem Namen ab und macht aus dem alten File ein Backup (.BAK)
SAVE AS	– ruft das Speichermenü auf

Leider wird nach dem Ausführen der Befehle SAVE und SAVE AS immer das Textfenster geschlossen, d. h. der Text wird gelöscht. Wenn man nur zwischenspeichern wollte, muß man den Text, zum Weiterarbeiten erst wieder einladen!

READ	– lädt ein Textfile und setzt an der momentanen Cursorposition ein
WRITE	– mit dieser Funktion kann man einen markierten Textbereich auf Diskette speichern
DELETE	– löscht ein Textfile auf Diskette
QUIT	– Verlassen des Textverarbeitungssystems
LAYOUT	– hier wird die Seiteneinteilung (vertikal) des Textfiles festgelegt, außerdem der Inhalt eventueller Kopf- oder Fußzeilen (siehe Ausdruck eines Textes)

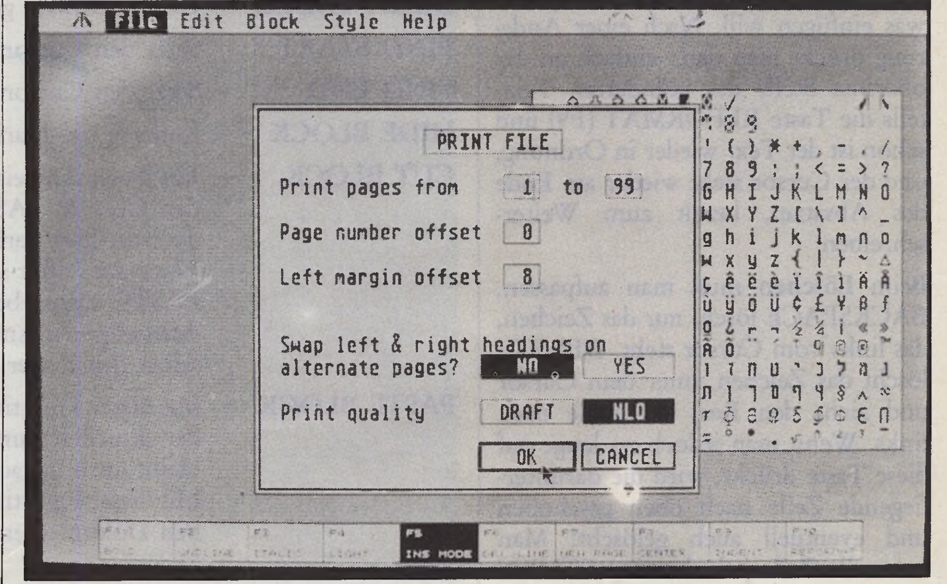


Bild 3: PRINT-Menü

3) Da das Programm von sich aus einen Seitenumbruch vornimmt (siehe Punkt 2) ist es manchmal sinnvoll, Textabschnitte, die nicht getrennt werden sollen, besonders zu markieren. Dazu geht man mit dem Mauszeiger auf den linken Balken (in dem die Seitenzahlen stehen) und drückt in Höhe der oberen Kante der zu markierenden Textstelle die Maustaste. Diese hält man nun gedrückt und fährt damit an das Ende der Textstelle. Der Bereich färbt sich dabei schwarz. Läßt man nun die Taste los, so wird der Bereich im Seitenbalken mit einer gestrichelten Linie gekennzeichnet. Wenn ein Seitenumbruch in diesen Bereich fallen sollte, dann wird er vor den so markierten Bereich gelegt.

Am oberen Rand des Textfensters befindet sich eine Linie, die den rechten Rand des Textes und die Tabulatormarken anzeigt. Zum Verschieben wird das Symbol für den rechten Rand mit der Maus angeklickt und dann (Taste gedrückt halten!) verschoben. Die Tabulatormarken können durch anklicken gesetzt oder gelöscht werden.

Editieren eines Textes

Es empfiehlt sich, beim **Editieren** im INS MODE (F5) zu sein, außerdem sollte man die Funktionen WORD WRAP und JUSTIFY eingeschaltet haben. Damit kann man gut Texte eingeben und sie auch jederzeit ändern, wenn man Fehler entdeckt hat oder etwas einfügen will. Nach einer Änderung drückt man dann einfach an der obersten Stelle des geänderten Textteils die Taste REFORMAT (F9) und schon ist der Text wieder in Ordnung, und der Cursor steht wieder am Ende des Absatzes, bereit zum Weiterschreiben.

Beim **Löschen** muß man aufpassen. BACKSPACE löscht nur das Zeichen, das links vom Cursor steht. DELETE löscht das Zeichen unter dem Cursor und zieht den Rest der Zeile nach links. Wenn man jedoch zu lange auf diese Taste drückt, wird die darunterliegende Zeile nach oben geschoben und eventuell auch gelöscht! Man kann die Zeile jedoch mit RETURN wieder nach unten bringen.

Das EDIT-Menü

WP MODE – im Zustand 'off' wird reiner ASCII-Code erzeugt. Dies ermöglicht das Erstellen von Quelltexten für Compilersprachen (Pascal, C usw.). Dadurch wird ein komfortableres Editieren ermöglicht.
Schaltet man den WP Mode an, so können alle Schrifttypen von 1ST WORD genutzt werden. Diese Funktion aktiviert alle Punkte des **Style**-Menüs.

INSERT MODE – damit wird der Einfügemodus eingeschaltet, das heißt, daß beim nachträglichen Einsetzen von Text keine Zeichen überschrieben werden können

FIND – es kann nach einem Wort oder Textteil gesucht werden, dabei kann auf Groß/Kleinschreibung geachtet (MATCH) oder dies außer Acht gelassen werden (IGNORE)

REPEAT FIND – mit dieser Funktion wird nach Auffinden der ersten Stelle nach einer weiteren gesucht

REPLACE – es wird nach einem Ausdruck gesucht, und dieser wird durch einen zweiten ersetzt, außerdem kann gewählt werden, ob ein, manche oder alle Ausdrücke ersetzt werden sollen

SET MARK – es können bis zu vier Marken gesetzt werden

GOTO MARK – springt zu der mit SET gesetzten Marke

Das BLOCK-MENÜ

START BLOCK – Hiermit kann der Anfang eines Blocks markiert werden (momentane Cursorposition).

END BLOCK – Markiert entsprechend das Ende eines Blocks.

Ein Block kann auch mit der Maus markiert werden. Man klickt den oberen Rand des Blocks an, hält die Maustaste gedrückt und fährt nach unten bis ans Blockende. Der Block ist nun genau wie oben markiert und die folgenden Funktionen können jetzt auf ihn angewendet werden:

COPY BLOCK – Der Block erscheint noch einmal an der neuen Cursorposition, er bleibt auch an der alten Stelle erhalten.

MOVE BLOCK – Der Block wird zur Cursorposition bewegt und dort eingesetzt, an der alten Stelle wird er gelöscht.

DELET BLOCK – Der Block wird gelöscht, die leere Stelle wird aufgerückt.

FIND START – Setzt den Cursor in die erste Reihe des Blocks.

FIND END – Setzt den Cursor unter das Ende des Blocks.

HIDE BLOCK – Entfernt die Markierung eines Textblocks.

CUT BLOCK – Mit dieser Anweisung wird ein markierter Textbereich in den CUT & PASTE-Puffer abgelegt. Dabei geht jedoch die Schriftart verloren.
Wenn im Puffer schon ein Text steht, der noch nicht mit PASTE ausgegeben wurde, erscheint eine Warnung.
Achtung: die Anweisungen COPY/MOVE BLOCK löschen den Puffer.

PASTE BLOCK – Mit dieser Funktion kann der im Puffer stehende Text an der aktuellen Cursorposition eingefügt werden. Der Text bleibt auch danach erhalten.
Mit dieser Funktion kann auch Text zwischen verschiedenen Dokumenten ausgetauscht werden.

Das STYLE-Menü

BOLD	– Schaltet den Fettdruck ein (auch über F1).
UNDERLINE	– Der Text wird ab dieser Stelle unterstrichen (F2).
ITALIC	– Aktiviert die Schrägschrift/Kursivschrift (F3).
LIGHT	– Erzeugt die hellgraue Schrift.
SUPER	– Superscript: hochgestellte Zeichen für Formeln oder als Kleinstschrift.
SUBSCRIPT	– Tiefgestellte Zeichen für chem. Formeln oder als Kleinstschrift.

Diese Schriftarten können untereinander gemischt werden, außer Fettschrift mit der hellgrauen Schrift.

RESTYLE	– Mit dieser Anweisung kann in einem markierten Textbereich der Schrifttyp geändert werden. Achtung, dabei verschwinden Schriftartänderungen in diesem Textbereich!
JUSTIFY	– Nach einem Wortumbruch wird die Zeile mit Leerzeichen aufgefüllt, bis sie rechtsbündig ist.
WORD WRAP	– Wenn diese Funktion eingeschaltet ist wird ein Wort, das über die Markierung des rechten Randes hinausgeht in die nächste Zeile gesetzt.

Wenn man verhindern will, daß beim Wortumbruch (WORD WRAP) oder beim Randausgleich (JUSTIFY) ein Ausdruck, der ein Leerzeichen, enthält (z. B. 1st_Word), getrennt wird, dann drückt man zusammen mit der SPACE-Taste noch auf die CONTROL-Taste.

SPACING	– Dieser Befehl setzt in einem markierten Bereich zwischen jeder Zeile noch eine Leerzeile, bzw. er löscht aus diesem Bereich die Leerzeilen. Der Befehl ist erst ab der Version 1.04 vorhanden.
CENTER	– Nach Drücken dieser Taste wird das in dieser Zeile stehende Wort oder Satzteil in der Mitte zentriert (F8).
INDENT	– Mit dieser Funktion kann der Text links eingerückt geschrieben werden (F9). Die Funktion kann an **allen** Tabulatormarken positioniert werden. Beim Wortumbruch wird diese Stelle dann als Textanfang erkannt.
REFORMAT	– Wenn man einen Text nachträglich ändert, so wird beim Einfügen über den rechten Rand geschrieben – mit dieser Funktion kann nun der Text wieder links/rechtsbündig angeordnet werden. Dazu bewegt man den Cursor, am besten mit der Maus, in die oberste geänderte Zeile und drückt die Taste REFORMAT (F10). Der Text wird dann auf das gegebene Format gebracht, wobei auch eingerückte Textabschnitte berücksichtigt werden.

Mit der Tastenkombination CONTROL & DELETE wird eine Zeile von der Cursorposition bis zum Ende gelöscht. Die Taste F6 löscht eine ganze Zeile. Man darf jedoch nicht vergessen, daß sie mit einer Repeat-Funktion ausgestattet ist und so im Nu ganze Textpassagen löschen kann. Die Anweisung DELETE BLOCK im Block-Menü löscht einen markierten Block.

Ausdrucken von Texten

Zum Ausdrucken eines Textes muß zuerst das Textfenster geschlossen werden, danach wählt man im File-Menü die Option **PRINT**. Es erscheint das bekannte Lade- bzw. Speichermenü, in dem der zu druckende Text eingegeben angibt.

Nun erscheint das **PRINT FILE** Menü (siehe Bild 3). Hier werden die Seitennummern eingegeben, die ausgedruckt werden sollen. Außerdem können alle Seitenzahlennummern um einen bestimmten Wert erhöht werden (Page number offset). Mit der nächsten Anweisung (Left margin offset) wird angegeben, wie weit der linke Rand beim Ausdruck eingerückt wird.

Mit der Funktion 'Swap...' wird der Text, der rechts und links in der Kopfzeile (siehe Layout) ausgedruckt werden soll, bei geraden Seitenzahlen vertauscht.

Mit der letzten Option in diesem Menü läßt sich nun noch die Near Letter Quality einstellen.

Die Kopf- und Fußzeile jedes Blattes läßt sich mit einem Text beschreiben. Dazu wählt man im Edit-Menü die Funktion **Layout** (siehe Bild 4). Nun kann in die vorgegebenen Zeilen der gewünschte Text eingegeben werden. Bei Eingabe von # erscheint dort die Seitenzahl. Den Ort dieser Textzeilen kann in diesem Menü auch festgelegt werden (Head/Foot margin usw.). Wenn man hiermit ein wenig experimentiert, kommt man schnell hinter die Bedeutung der einzelnen Punkte.

Beim Ausdrucken kann der **Drucker angehalten** werden, wenn man die Maustaste eine Zeit lang drückt. Durch erneutes Anklicken des Print-Fensters wird das Ausdrucken fortgesetzt.

Druckeranpassung

Eine weitere erfreuliche Fähigkeit von 1st Word ist die komfortable Druckeranpassung, die den Anschluß eines jeden Druckers ermöglicht.

Für die folgenden Drucker wird die Anpassung schon als ★.HEX mitgeliefert:

1. Matrixdrucker:
   EPSON RX/FX-80
   EPSON LX (NLQ)
   ATARI SMM804
   ASCII-DRUCKER
2. Typenraddrucker:
   QUME
   BROTHER HR-15/25

Bei diesen Druckern muß nur noch wie unten beschrieben (ab Punkt 7) verfahren werden.

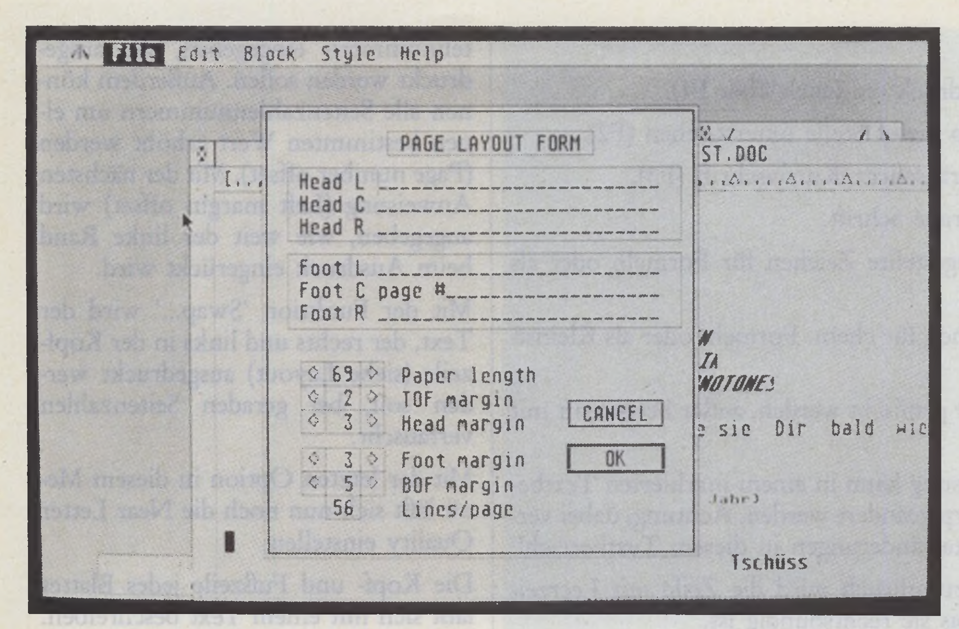


Bild 4: LAYOUT-Menü

Jeder Code, der vom Rechner zum Drucker geschickt wird kann auf den vorhandenen Drucker umgesetzt werden. Somit ist ein problemloses Anpassen jeglicher Schriftarten, Sonderzeichen und sonstigen Druckersteuercodes möglich. Ebenso kann ein Grafikzeichen, das nicht im Druckerzeichensatz enthalten ist, durch Übergabe der betreffenden Bit-Muster im Bit-Map Modus auf dem Drucker ausgegeben werden. Dies ermöglicht den Ausdruck des gesamten Zeichensatzes (siehe Bild 5) auf allen grafikfähigen Matrixdruckern.

Bei der 1st Word Version 1.04 ist bereits ein File **LESE MICH** enthalten, in dem (auf deutsch) erklärt wird, wie man sich eine Anpassung an den eigenen Drucker erstellen kann. Außerdem ist hier auch angegeben, wie der, vom Drucker am Anfang gesendete, Seitenvorschub (nur bei der alten Version 1.00) in einen Zeilenvorschub geändert wird:

a) EPS__RX80.HEX und
EPSON-Kompatible:
statt 1E,C
neu 1E,D

b) ASCII.HEX
(IBM und Kompatible):
statt ★1E
neu 1E,0

Für Besitzer der 1ST WORD Version 1.00 folgt nun in kurzer Form die Anleitung zum Erstellen eines geeigneten Druckertreibers:

1. Laden Sie 1ST WORD.PRG
2. Laden Sie einen der Files mit dem Zusatz .HEX, der Ihren Druckercodes am ähnlichsten ist (siehe Anleitung für Ihren Drucker).
(Um sich einen Überblick zu verschaffen, sollten Sie einmal probieren, wie Ihr Drucker auf den voreingestellten Druckertreiber reagiert. Gehen Sie im FILE-Menü auf den Punkt PRINT... und lassen Sie sich den Text ausdrucken.)
3. Schalten Sie im Menü EDIT den WP MODE aus.
4. Ändern Sie in diesem Text die Druckercodes entsprechend Ihrem Druckerhandbuch (meist sind es nur wenige, die nicht übereinstimmen). (Ein ★ vor der Zeile bedeutet, daß diese Zeile nicht aktiviert ist)
5. Speichern Sie das geänderte Programm ab.
6. Verlassen Sie 1ST WORD und laden Sie INSTALL.PRG (im Ordner PRINTER).
7. Laden Sie nun den von Ihnen geänderten Textfile (.HEX).
8. Nun wird ★.DOT bzw. ★.DSY erstellt.

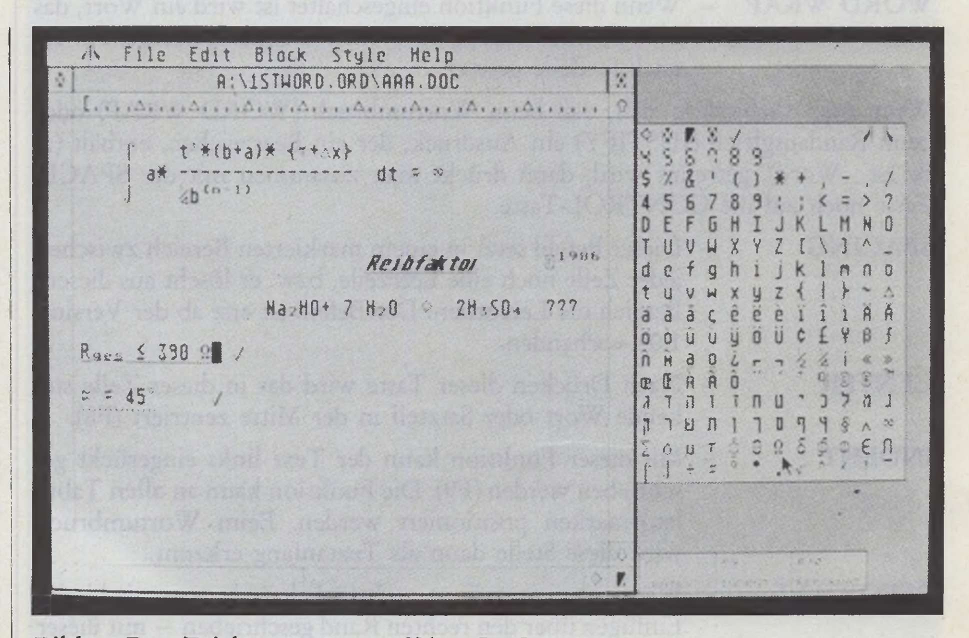


Bild 5: Der Zeichensatz von 1ST WORD

9. Das entstehende Programm müssen Sie jetzt aus dem PRINTER-Ordner in das Hauptprogramm kopieren. Der dort stehende alte Druckertreiber ★.DOT bzw. ★.DSY muß deshalb gelöscht werden.

Kleine Mängel am Programm

– 1st Word verfügt über keine Worttrennung, was bei einigen Fällen von Nachteil sein kann. Bei langen Wörtern kommt es dadurch zu großen Zwischenräumen in den Zeilen.
– Spaltenverarbeitung über die beispielsweise der ST-MANAGER TEXT verfügt, ist bei 1st Word

nicht vorgesehen.

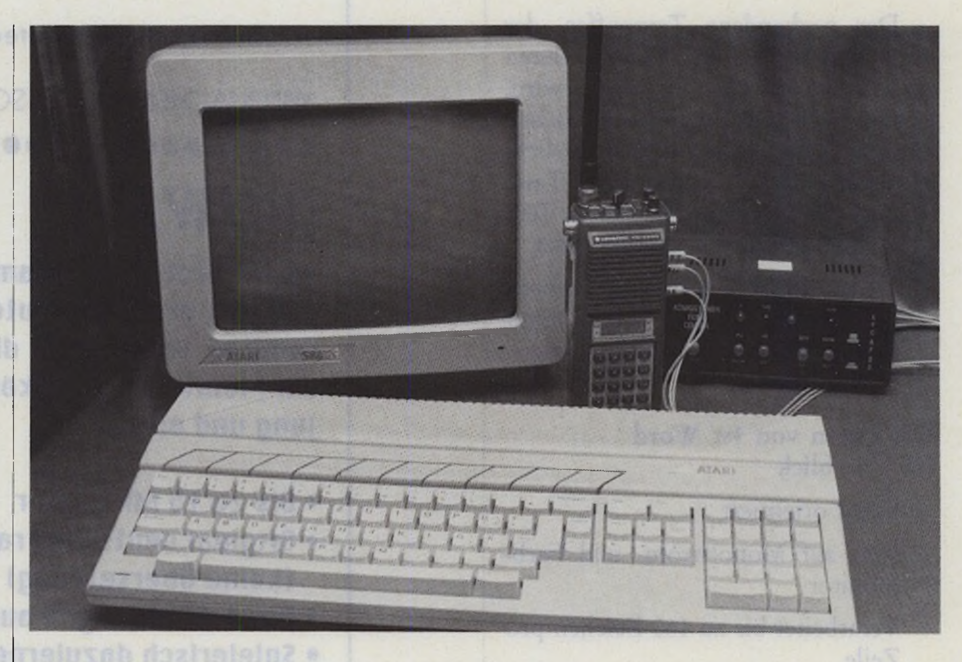
- Der vorhandene Textpuffer, der beim Kopieren und Transportieren von Textpassagen eingesetzt wird, tritt beim Löschen nicht in Erscheinung. Dies ist besonders ärgerlich, wenn man versehentlich die Taste F6 (DEL LINE) drückt und der Text unwiederbringlich verschwindet.
- Leider ist die Funktionstastenbelegung festgelegt und kann somit nicht den eigenen Wünschen angepaßt werden.

Die Daten von 1st Word im Überblick

- GEM orientiert
- läuft auf Monochrom- und Farbmonitor
- Verarbeitet bis zu 160 Zeichen pro Zeile
- Fett-, Hell-, Kursivschrift, Super-, Subscript und Unterstreichen sind sowohl auf dem Monitor als auch mit dem Drucker darstellbar
- beherrscht den gesamten ATARI-Zeichensatz
- bis zu vier Dokumente können sich gleichzeitig im Speicher befinden und abwechselnd bearbeitet werden
- Textaustausch zwischen den Dokumenten ist möglich
- Speichern und Laden von Textteilen
- Seitenumbruch (automatisch und manuell)
- Randausgleich (links und rechts), Zentrieren
- Tabulatorfunktionen (TAB, INDENT)
- Funktionen zum Suchen und Ersetzen
- komfortable Druckeranpassung
- Maussteuerung
- Layout, Tabulatorzeile wird abgespeichert
- optional Mail Merge (1st Mail)

Der ATARI ST wird zum RTTY-Terminal

Die Funkamateure können aufatmen. Hier wird das erste Programm vorgestellt, das unter Zuhilfenahme einfacher Hardware, einem sogenannten Konverter, den Sende- und Empfangsbetrieb in der Betriebsart RTTY (Radio Teletype – Funkfernschreiben) ermöglicht.



Gerade in den letzten Jahren hat das Interesse am Funkfernschreiben deutlich zugenommen. Das hängt wohl auch damit zusammen, daß sich ein Computer sehr gut für dieses Hobby eignet. Mußte man sich noch vor einigen Jahren einen Fernschreiber zulegen, so genügt es heute, einen Computer mit dem entsprechenden Programm zu „füttern". Funkfernschreiben ist somit ein Musterbeispiel dafür, wie sich früher rein hardwaremäßige Aufgaben heute mit Software bewerkstelligen lassen. Das hier vorgestellte RTTY-Programm der Firma KFC aus Königstein schließt nun auch diese Softwarelücke des ATARI ST.

Das Programm wurde von DG6FH in der Programmiersprache 'C' geschrieben und arbeitet auf TOS-Ebene, d. h. die gesamte Bedienung erfolgt über die Tastatur und nicht mit der Maus. Ob dies bei einem Programm, welches ausschließlich mit Texten arbeitet, sinnvoll ist, darüber kann man streiten. Sicher wäre ein häufiges Wechseln zwischen Tastatur und Maus auf die Dauer eher lästig, andererseits wäre die Fenstertechnik oder „Helpmenüs" auf GEM-Ebene an manchen Stellen wünschenswert. Das Programm arbeitet sowohl mit dem monochromen, hochauflösenden Monitor, als auch mit einem Farbmonitor in mittlerer Auflösung. Bevor Sie das Programm starten, sollten Sie die Uhr im Kontrollfeld auf GMT (Weltzeit) einstellen, da es im Programm nicht mehr möglich ist. Damit die Daten übertragen werden können, muß selbstverständlich ein Konverter vorhanden sein, der die Töne vom Empfänger für „Mark" und „Space" in computergerechte Signale umwandelt und umgekehrt die Rechteckimpulse des Rechners in Töne, die gesendet werden können. Ein solcher Konverter, der übrigens auch die Sende-Empfangsumschaltung der Funkstation (PTT) steuert, wird bei diesem Programm an den Druckerport angeschlossen.

Nach dem Starten (Doppelklick auf RTTY.TOS) ist der Rechner mit der – im Amateurfunk üblichen Geschwindigkeit von 45 Baud – empfangsbereit. Dies wird durch ein 'RX:' auf dem Bildschirm signalisiert. Nun kann man, wenn es gewünscht wird, die Geschwindigkeit ändern (siehe Tabelle), und zwar für Senden und Empfang getrennt. Dies ermöglicht es, z. B. mit 100 Baud zu senden und mit 45 Baud zu empfangen. Das Wechseln der Geschwindigkeit ist auch während des Empfangs möglich.

Tabelle der möglichen Geschwindigkeiten

F1	senden	45 Baud
F2	senden	50 Baud
F3	senden	57 Baud
F4	senden	75 Baud
F5	senden	100 Baud
F6	empfangen	45 Baud
F7	empfangen	50 Baud
F8	empfangen	57 Baud
F9	empfangen	75 Baud
F10	empfangen	100 Baud

Damit sind alle Geschwindigkeiten, bei denen der Baudot-Code benutzt wird, vorhanden. Denn bei größerem Tempo ist dieser Übertraungscode zu fehleranfällig, und es wird der ASCII-Code verwendet. Diesen kann das Programm allerdings nicht interpretieren. Funkamateure, die mit dem ASCII-Code arbeiten möchten, können auf die RS 232 Schnittstelle und das VT 52 Terminalprogramm, welches von ATARI mitgeliefert wird, zurückgreifen. Ein Grund für die Fehleranfälligkeit des Baudot-Codes ist, daß sowohl für Buchstaben als auch für Zeichen und Ziffern die gleiche Bitfolge (Zeichenfolge) benutzt wird, nur mit dem Unterschied eines vorangestellten Umschaltbefehls. Geht nun dieser Umschaltbefehl durch eine Störung verloren, so schreibt der Computer z. B. weiter wirre Zahlen und Satzzeichen auf den Bildschirm, obwohl es längst wieder Buchstaben sein sollten. Das Programm bietet hier die Möglichkeit, durch Drücken der 'Help'-Taste die-

sen Umschaltbefehl auch auf der Empfängerseite manuell zu geben, so daß der Rechner wieder Buchstaben schreibt. Zur Störsicherheit des Programms ist noch festzustellen, daß durch den hohen Systemtakt des ST jedes empfangene Zeichen (samt Startbit) ca. 20 bis 50 mal, je nach eingestellter Geschwindigkeit, geprüft wird, so daß ein Störimpuls, der kleiner als 22 Millisekunden ist, erkannt und ignoriert wird.

Die Sende- Empfangsumschaltung erfolgt durch Drücken der Taste '★'. Danach schaltet der Rechner mittels der PTT-Leitung den Sender ein und wartet auf eine Eingabe der Tastatur. Ein 'TX:' kennzeichnet den Sende-Modus. Gut gelöst wurde das Aussenden von Standardtexten, wie z. B. CQ-Rufe, Stationsbeschreibung, etc.. Gab es bisher nur die Möglichkeit, solche vorbereiteten Texte fest im Programm und somit nur schwer änderbar unterzubringen oder aber im RAM, wo sie nach jedem Ausschalten des Rechners gelöscht werden, so werden Ihre Texte hier als ASCII-File auf Diskette geschrieben. Damit sind diese Texte jederzeit leicht änderbar und trotzdem dauerhaft auf der Systemdiskette gespeichert. Außerdem sind sie in ihrer Länge nur durch die Diskettenkapazität beschränkt, was wohl in jedem Fall ausreichen dürfte. Diese Texte (ASCII-Files) können mit jedem Editor bzw. Textverarbeitungsprogramm erstellt werden. Mitgeliefert wird der Editor ST-WRITER. Den Texten kann man einen beliebigen Namen geben, der allerdings auf '.F<n>' enden muß. Dabei steht <n> nur als Platzhalter für eine Zahl zwischen 1 und 10, welche später der Funktionstaste entspricht, mit der der Text im Sende-Modus aufgerufen wird. So wird z. B. ein Text-File mit dem Namen CQCQCQ.F1 mit der Funktionstaste F1 aufgerufen und automatisch gesendet. Das Programm sendet keinen automatischen Zeilenvorschub / Wagenrücklauf, erinnert jedoch nach 54 Zeichen durch eine Glocke daran. Nach 64 Zeichen ertönt die Glocke dann nach jedem Tastendruck. Durch Drücken der 'Return'- bzw. 'Enter'-Taste wird ein Wagenrücklauf **und** Zeilenvorschub gesendet. Betätigt man nochmals die Taste '★', wird der Sende-Modus verlassen und auf Empfang umgeschaltet. War der gesendete Text jedoch länger als 30 Zeichen, wird automatisch noch das eigene Rufzeichen und die Uhrzeit mit ausgesendet (siehe Bild 2). Diese Tatsache soll den lizensierten Funkamateur vor Störern bzw. Raubkopierern schützen. Da das eigene Rufzeichen fest ins Programm eingebaut ist, muß es bei Bestellung mit angegeben werden.

Bewertung

Mit dem Program läßt sich vernünftig arbeiten, obwohl es in seiner Gesamtstruktur einfach gehalten ist. Sehr angenehm ist die Möglichkeit, vorbereitete Texte oder auch ganze Programme von der Diskette her zu senden. An Luxusausstattung wurde jedoch gespart, so wäre ein geteilter Bildschirm für Empfang und Senden wünchenswert. Angenehm wäre auch die Möglichkeit, den empfangenen Text gleichzeitig abspeichern oder drucken zu können. Eine Statuszeile, die Uhrzeit, Übertragungsgeschwindigkeit, oder auch die Funktionstastenbelegung, die ja bei Empfang und Senden unterschiedlich ist, ständig anzeigt, würde die Übersichtlichkeit erhöhen. Andere Betriebsarten wie AMTOR oder CW sind natürlich enorm hohe Ansprüche, die das Programm (noch) nicht erfüllen kann.

Wie uns die Firma KFC mitteilte, wird an einer verbesserten Version, die verschiedene Fenster für Empfang und Senden zur Verfügung stellt, gearbeitet. Diese ist dann gegen einen Aufpreis und Rücksendung der Original-Diskette zu erhalten. Das hier getestete RTTY-Programm kostet 98,– DM incl. MwSt. und kann unter folgender Adresse bezogen werden:

KFC-Computer
Wiesenstr. 18
6240 Königstein
Tel. 06174/3033 (UB)

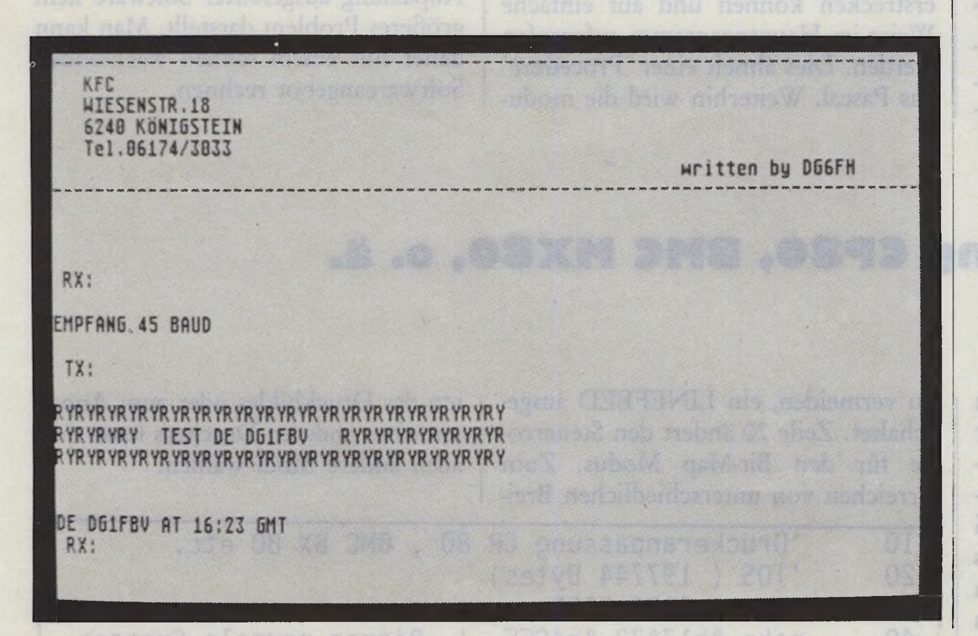


Bild 2: Bildschirm nach dem Einschalten

Basic-Compiler

PHILON FAST/BASIC

Kurztest

Kurz nach Redaktionsschluß erreichte uns eine Test-Version des Basic Compilers von PHILON Inc, einer amerikanischen Softwarefirma, die neben diesem Compiler auch die Sprachen Fortran, Cobol, Pascal und C in Ihrem Programm hat. Der Basic-Compiler wird auf fünf Disketten mit umfangreichem Handbuch geliefert. Um es vorweg zu sagen, dieser Compiler verfügt über einen eigenen Sprachumfang und hat somit nichts mit dem ST-Basic zu tun. Die fünf Disketten enthalten u. a. folgende Programme:

- Basic Compiler
- Code Optimierungsprogramm
- Assembler
- Assembler Optimierungsprogramm
- Linker
- Editor
- BC + IEEE Mathematikbibliothek
- Grafikbibliothek

Der Assembler ist auch einzeln einsetzbar. Den Assembler-Quelltext kann man mit dem beigefügten Editor erstellen. Der Editor (Mark Williams) verfügt über vielfältige Editiermöglichkeiten, die allerdings nur über Control-Sequenzen erreichbar sind.

Besonderheiten

Dieser Compiler beherrscht den Sprachumfang des bekannten 'MBASIC' von Microsoft. Er ist für mathematische Anwendung geradezu geschaffen. Die Rechengenauigkeit aller mathematischen Funktionen beträgt ein Vielfaches derer des ST-Basics. Dazu einige Daten:

Integer:

dezimal	– 2 147 483 648	bis 2 147 483 647
oktal	– 20 000 000 000	bis 17 777 777 777
Hexadezimal	– 80 000 000	bis 7F FFF FFF

Real:

BCD : 14 Stellen E + –64
IEEE : 16 Stellen E + –308

Wie diese Tabelle zeigt, sind alle mathematischen Funktionen in zwei verschiedenen Rechenformaten ansprechbar. Das BCD-Format legt intern die Zahlen dezimal ab, wodurch die sonst bei Computern auftretenden Rundungsfehler ausgeschlossen werden. Das andere Format stellt die Zahlen im sonst rechnerüblichen dualen Format dar und drechnet daher mit höherer Geschwindigkeit.

Im Gegensatz zu sonstigen Basic-Versionen bietet dieser Compiler eine erweiterte 'DEFFN'-Funktion. Sie gestatten das Erstellen komplexer Funktionen, die sich über mehrere Zehlen erstrecken können und auf einfache Weise im Hauptprogramm aufgerufen werden. Dies ähnelt einer 'Procedure' aus Pascal. Weiterhin wird die modulare Programmierung unterstützt. Einerseits können Programmteile, ähnlich einem MAKRO in Assembler, an eine bestimmte STelle des Hauptprogramms eingefügt werden, andererseits ist es auch möglich vorher compilierte Programme sowie Assemblerprogramme aufzurufen.

Grafikbefehle sind nicht im Sprachumfang vorhanden, grundsätzlich ist aber das Einbinden von solchen möglich.

Der Betrieb des Compilers ist für zwei Laufwerke oder eine Festplatte vorgesehen.

Am Beispiel dieses Compilers, der ursprünglich für andere Systeme entwickelt wurde, zeigt sich, daß eine Anpassung ausgereifter Software kein größeres Problem darstellt. Man kann daher mit einem schnell wachsenden Softwareangebot rechnen.

Druckeranpassung CP80, BMC MX80, o. ä.

Gültig für TOS 197744 Bytes

Für die Anpassung eines solchen Druckers ergaben sich auf Grund der neuen TOS-Version einige Änderungen. Die Tabelle der Druckersteuersequenzen befinden sich bei diesem TOS im Rechnerspeicher von $17A50 bis $17A90. In Zeile 10 wird, um einen kleineren Spalt zwischen den Zeilen zu vermeiden, ein LINEFEED ausgeschaltet. Zeile 20 ändert den Steuercode für den Bit-Map Modus. Zum Erreichen von unterschiedlichen Breiten des Druckbildes oder zum Anpassen eines anderen Druckers kann man auch andere Modi wählen.

```
10    'Druckeranpassung CP 80 , BMC BX 80 etc.
20    'TOS ( 197744 Bytes)
30    poke &h9D56,0000
40    poke &h17A72,&h4CFF  ' Bitmap normale Groesse
```

Software für ATARI 260/520 ST

Ab jetzt können Sie Ihren ST noch besser nutzen!

7 preiswerte, extrem leistungsfähige Programmpakete für Ihren ST. Und so wird bestellt: Bestellen Sie eine Position, zahlen Sie den angegebenen Preis. Alle 7 Pakete zusammen plus die drei Bücher nur **DM 499,–. Sie sparen DM 82,60.**



ST-Adress
Eine sehr leistungsfähige Adressenverwaltung für Ihren ATARI 520 ST / 260 ST. Ca. 2000 Adressen pro Diskette, ca. 40000 Adressen bei Festplatte. Selektieren nach allen Feldern möglich, die beiden ersten Felder mit Stichwort. Suchmöglichkeit, sehr einfache Bedienerführung, Aufkleber und Listen drucken, sehr schnell auch bei großen Datenmengen.
Best.-Nr. 7405 — DM 49,–

68000 Programmier-Handbuch
Best.-Nr. 25 — DM 39,– Buch

16-Bit-Microcomputer, Einführung – Anwendungen
Best.-Nr. 116, 373 Seiten — DM 29,80

MODULA-2 Anwender-Handbuch
Best.-Nr. 223 — DM 29,80



ST-Adress Notiz
In den Leistungsmerkmalen wie ST-Adress, jedoch kann jeder Adresse ein Notizfeld zugeordnet werden. Ideal für alle, die Adressen verwalten und sich zu jeder Adresse ein paar Notizen machen wollen. Zwei Versionen sind im Lieferumfang enthalten. Kleines Notizfeld mit ca. 120 Zeichen Platz für Notizen pro Adresse und eine Version mit 896 Zeichen Platz für Notizen. Komfortable Bildschirmanzeige und Druckerausgabemöglichkeiten sind eingebaut. Unzählig viele Anwendungen möglich.
Best.-Nr. 7400 — DM 79,–

ST-Auftragsbearbeitung
Integrierte Lager- und Adressenverwaltung mit Fakturierung. Lagerverwaltung: sehr komfortabel, Mindestmengen, Bewertung nach Ein- und Verkaufspreisen. Adressenverwaltung: Suchen, Aufkleber und Listen drucken. Fakturierung: Rabatte automatisch oder von Hand, MwSt., Fracht etc. alles frei wählbar, schreibt Aufkleber u.v.a. Adresse und Produkt können aus Dateien aufgerufen werden, verkaufte Produkte werden automatisch vom Lager abgebucht. Sie erhalten den gesamten Quelltext in BASIC, so daß Sie auch evtl. selbst erweitern können.
Best.-Nr. 7406 — DM 99,–



ST-Rundschreiben
Erstellung von Serienbriefen mit integrierter Adressenverwaltung. Das ist eine echte Preis-Sensation! Mit der Textverarbeitung kann ein Text oder ein Brief erstellt werden. Dieser kann dann an alle oder an ausgesuchte Adressen aus der integrierten Adressenverwaltung verschickt werden. Das Konzept ist so angelegt, daß ein Benutzer mit einem Minimum an Befehlen das Programm bedienen kann. Kurze Einarbeitungszeiten bei hohem Nutzen sind möglich. Selektieren, Suchen, Aufkleber ausdrucken, Breit- und Fettdruck im Text, Text kann gespeichert werden, sind nur einige der vielen Möglichkeiten dieses Paketes.
Best.-Nr. 7404 — Diskette mit ausf. Anl. DM 79,–

Weitere nützliche Bücher für Ihren ST

Best.-Nr.	Titel	Preis
102	Mathematische und wissensch. Programme	DM 29,80
188	Statistik in BASIC	DM 39,00
194	C-Handbuch, Einführung und Beispiele	DM 29,80
220	Tabellenkalkulation Einf. für Kuma + VIP	DM 19,80
216	WordStar für Füchse	DM 24,80

Die Best.-Nr. 7406 und Best.-Nr. 7407 können jetzt schon bestellt werden, werden aber erst geliefert, wenn eine endgültige BASIC-Version zur Verfügung steht. Alle anderen Produkte sind sofort lieferbar. Bis eine endgültige Version von GEM verfügbar ist, werden die Programme als Tastatur-Versionen geliefert. Ein kostenloser »Upgrade« ist später für alle rechtmäßigen Besitzer durch Einsendung der Original-Diskette mit Rückporto möglich.



ST-Karteikarten-System
Dieses Datenbank-System leistet mehr als eine gewöhnliche Datenverwaltung. Hier wird einfach das Prinzip der Karteikarte auf Ihren ST übertragen. Im Lieferumfang sind drei »Karteikarten« verschiedener Größe enthalten. 256, 512 und 1024 Zeichen pro Karteikarte. Innerhalb der Karteikarte können beliebige Eintragungen gemacht werden, nach denen gesucht oder selektiert werden kann. Jedes Zeichen oder jeder beliebige Ausdruck kann Schlüsselwort sein. Eine logische UND-Verknüpfung ist möglich. Sehr leicht zu bedienen. Bis zu 1000 Karteikarten pro Diskette, bei Festplatten-Versionen entspr. mehr. Sehr schnelles Suchen möglich. Unzählige Anwendungen in fast allen Bereichen, Ärzte, Rechtsanwälte, Autohandel, Immobilien, Makler etc.
Best.-Nr. 7402 — Diskette mit ausführlicher Anleitung DM 79,–



ST-Lager
Lagerverwaltung. Sehr komfortable Bedienerführung. Mindestmengen, Lagerwerte ermitteln, Hersteller und Produktcodierung, Letzter Lagerzugriff für die Ermittlung von Schleichern und Rennern. Sie erhalten den gesamten Quelltext und können so leicht ändern und anpassen.
Best.-Nr. 7407 — DM 49,–



ST-Literaturverzeichnis
Das ideale Programm zum Abspeichern und Suchen von Zeitschriften-Artikeln, Gerichtsurteilen, Bücher, Archivierung usw. 512 Zeichen pro Feld sind möglich, nach denen stichwortartig gesucht werden kann.
Best.-Nr. 7401 — Diskette plus Anleitung DM 49,–



Ing. W. Hofacker GmbH
Tegernseer Str. 18
D-8150 Holzkirchen/Obb.
Tel. 08024/7331
Telex 526973

Bei NN + DM 6,50 NN-Gebühr.
Lieferung per NN oder Euroscheck oder Vorkasse, Pschk. Mnch. 15994-807 oder Eurocard oder American Express.

Für eilige Bestellungen ★ Bestell-Coupon

Heute noch ausfüllen und an Hofacker, D-8150 Holzkirchen, absenden!
Bitte senden Sie mir folgende Best.-Nr. per NN, Vork., Pschk, Euroscheck liegt bei,
Eurocard-Nr. .... Exp. Date ....
American Express-Card-Nr. .... Exp. Date ....
Gewünschte Nr. einfach ankreuzen:
7405 ★ 7400 ★ 7406 ★ 7404 ★ 7402 ★ 7407 ★ 7401 ★ 25 ★ 116 ★ 223
Ich wünsche alle Produkte zusammen zum Preis von DM 499,–. Sie sparen DM 82,60! Ja/Nein
Name ....
Straße .... PLZ u. Ort ....
Unterschrift ....

GEM-Routinen Teil 2

- Mausabfrage
- Definition des Maussymbols
- Markierungsbefehle
- weitere Grafikbefehle



Grundprinzip

Wie schon in der letzten Ausgabe der 'ST-Computer' beschrieben wurde, kann man mit Hilfe von GEM-Routinen viele nützliche Eigenschaften des ATARI ST erreichen, die leider im ST-Basic nicht enthalten sind. Der eigentliche Aufruf dieser Routinen erfolgt durch Angabe der jeweiligen Funktionsnummer und der dazugehörenden Parameter in den vordefinierten Feldern 'CONTRL, INTIN, PTSIN' mit anschließendem 'VDISYS'-Befehl.

Zur Erleichterung und zur Bewahrung des Überblicks wurden die GEM-Routinen in Unterprogramme eingebaut, die durch einen Namen, der dem Sinn der Routine entspricht, aufgerufen werden. Sämtliche Unterprogramme sind so aufgebaut, daß man sie alle gleichzeitig in einem Programm nutzen kann, deshalb sollten die Zeilennummern übernommen werden. Der Aufruf erfolgt wie bei jedem anderen Unterprogramm durch 'Gosub Programmname', jedoch müssen vorher die benötigten Variablenwerte übergeben werden.

Eine nützliche Eigenschaft der verwendeten GEM-Routinen ist, daß fast alle mit normalen Basicbefehlen kombinierbar sind.

Mausabfrage

Position & Tasten

Einer der wichtigsten Bestandteile des GEM, stellt die Maus dar, denn ohne dieses hilfreiche Tierchen wäre der ATARI ST wohl kaum noch denkbar. Denjenigen, die auch in Basic auf dieses Hilfsmittel nicht verzichten wollen, können auf einige GEM-Routinen zurückgreifen. Die interessanteste Routine ist die zur Abfrage der Mausposition und der Maustasten. Die dadurch entstehenden Möglichkeiten sind vielfältig, sie reichen von Eingabemenüs, bei denen die Abfrage der Menüpunkte nicht mehr mit Zahlen oder Buchstaben erfolgt, sondern durch einfaches Anklicken, bis hin zu grafischen Anwendungen, bei denen die Fähigkeiten dieser Routine erst zur Geltung kommen.

Der Aufruf dieses Unterprogrammes erfolgt durch:

Gosub Maus

```
63000 MAUS:
63002 ' <-- XMAUS,YMAUS,KEY
63004 poke contrl ,124
63006 vdisys
63008 xmaus =peek(ptsout)-1
63010 ymaus =peek(ptsout +2)-38
63012 key =peek(intout)
63014 return
63016 '--------------------
```

Routine Maus

```
5     ' merge "MAUS.bas"
9     '
10    fullw 2: clearw 2
20    color 1,1,1,1,1
30    start:    gosub maus
40    if key =1 then pcircle xmaus,ymaus,5,0,3600:
50    if key =2 then gotoxy 0,0 : print xmaus,ymaus
60    if key =3 then clearw 2
70    goto start
```

Listing 1: Maus demo

Die Routine liefert die gewünschten Werte in den Variablen 'XMAUS; YMAUS; KEY', wobei in XMAUS und YMAUS die Koordinaten der Maus enthalten sind.

Die Koordinaten sind in bezug auf das Output-Window normalisiert, da diese Routine, genau wie alle anderen VDI-Routinen, direkt den Bildschirm ansprechen. Die Variable 'KEY' enthält die Informationen über beide Maustasten, wobei 'Bit 0' bei der linken Taste und 'Bit 1' bei der rechten Taste eine Eins enthält. Somit sind drei Zustände abfragbar:

gedrückte Taste	Wert in 'KEY'
links	1
rechts	2
links + rechts	3

Ein einfaches Beispiel zu dieser Routine zeigt Listing 1.

Showmaus

Bei einigen Anwendungen kann es lästig werden, daß das Maussymbol (Grafikcursor) des öfteren nicht auf dem Bildschirm zu sehen ist, denn bei jeder Bildschirmausgabe oder bei der Tastatureingabe wird es unsichtbar. Erst wenn man die Maus bewegt, erscheint das Symbol wieder. Man kann es aber auch per GEM-Befehl sichtbar machen.

Bei dem hier angegebenen Unterprogramm geschieht dies durch:

Gosub SHOWMAUS

Entsprechend zu dieser GEM-Routine, existiert noch eine weitere zum Unsichtbarmachen des Grafikcursors (Funktionsnummer 123), die allerdings in Basic, außer eventuellen Blinkeffekten, keine bedeutende Anwendung findet.

```
63030 showmaus:
63032 poke contrl,122
63034 poke intin,0
63036 vdisys
63038 return
63040 '--------------
```

Routine SHOWMAUSE

Mausform

Der ATARI ST hat noch einige andere Maussymbole zur Verfügung, welche auch in Basic eingeschaltet werden können (Siehe Bild 1). Zur Realisierung dieser Funktion wurde ein kleiner Abstecher in die AES-Programmierung unternommen. Auf das Prinzip dieser Programmierung soll hier aber nicht näher eingegangen werden.

Der Aufruf erfolgt durch:

MAUSNR = 0 .. 7 :
Gosub MAUSFORM

Form	Wert
Pfeil	0
Cursorsymbol	1
Biene	2
zeigende Hand	3
Hand	4
dünnes Kreuz	5
dickes Kreuz	6
hohles Kreuz	7

```
63280 MAUSFORM:
63282 '--> MAUSNR
63284 '     0-7
63286 if mausnr >7 then return
63288 add#=gb
63290 gintin=peek(add#+8)
63292 addrin=peek(add#+16)
63294 poke gintin,mausnr
63296 poke addrin,0
63298 gemsys(78)
63300 return
63302 '-------------------------
```

Routine MAUSFORM

Definieren des Maussymbols

Wem diese sechs zusätzlichen Maussymbole nicht genügen, dem steht die Möglichkeit offen, eigene Symbole zu entwerfen. Bei manchen Programmen, die auf die Maus zurückgreifen ist es sinnvoll, wenn das Maussymbol etwas über die momentane Aufgabe des Programmes aussagt (z. B. Stift, Pinsel, Radiergummi).

Eine Möglichkeit zum freien Gestalten des Maussymbols bietet die VDI-Routine Nr. 111, die auch im folgenden Unterprogramm benutzt wird. Vor Aufruf dieser Routine muß das Muster des Maussymbols und das Hintergrundmuster, jeweils aus 16 Worten je 16-Bit bestehend, in bestimmten Speicherzellen übergeben werden.

Das Hintergrundmuster gibt an, welche Punkte des Feldes (16★16) unter dem Maussymbol sichtbar bleiben und welche generell gelöscht werden. Dies ist wichtig, wenn sich die Maus über dunkle Flächen bewegt. Es ist daher empfehlenswert, die Hintergrundmaske immer etwas größer zu wählen als das eigentliche Mausmuster.

In Speicherzellen 'INTIN +8' wird angegeben, ob das Muster schwarz oder weiß angezeigt wird, in Speicherzelle 'INTIN +6' die entsprechende Farbinformation des Hintergrundes.

'INTIN +2' und 'INTIN +4' legen den Punkt innerhalb des Mauszeigers fest, der bei der Positionsabfrage dem Grafikcursor entspricht. So sollte beispielsweise bei einem Pfeil dieser Punkt an der Pfeilspitze liegen.

Bei diesem Programm wurden verschiedene Maussymbole in Data-Zeilen gespeichert. Vor dem Lesen dieser Datas muß der Data-Zeiger auf die jeweiligen Datas gesetzt werden.

Der Aufruf erfolgt somit durch:

Restore Datazeile
Gosub MAUSDATA

Daraufhin erscheint das gewählte Maussymbol.

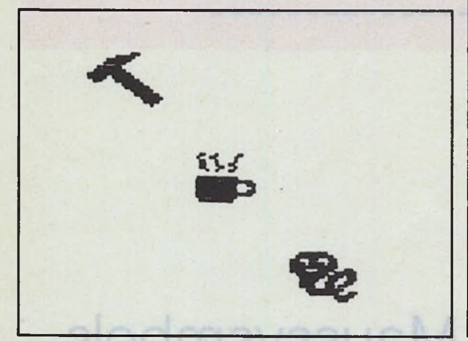


Bild 1

```
63140 '--------- MAUSDATAS -------------
63141 hammer:
63142 data 96,480,960,1984,3968,8064
63143 data 7936,16256,15872,32512,32512,65408
63144 data 65408,65472,63424,65504,25568,65520
63145 data 496,62456,248,508,124,254
63146 data 62,127,31,63,14,31
63147 data 4,14
63150 kaffee:
63151 data 4624,16184,9248,32376,4640,16368
63152 data 4672,16352,4384,16352,2624,8128
63153 data 0,16320,16352,32764,16380,32766
63154 data 16358,32767,16354,32767,16358,32767
63155 data 16380,32767,16352,32766,8128,16352
63156 data 0,8128
63160 wurm:
63161 data 0,8064,8064,16320,16320,32736
63162 data 26208,65520,30560,65520,32736,65532
63163 data 29132,65534,16318,32767,8054,16383
63164 data 7782,16383,7372,16382,7384,16380
63165 data 4080,8184,2022,4095,60,2046
63166 data 24,62
63199 '--------------------------------------
63200 mausdata:
63202 '--> restore zeile
63204 for a=0 to 15
63206 read mustervor,musterhinter
63208 poke intin +a*2+42 ,mustervor
63210 poke intin +a*2+10 ,musterhinter
63212 next
63214 '-----------------------
63250 mausneu:
63252 poke contrl,111
63254 poke contrl+6,37
63256 poke intin,5
63258 poke intin+2,5
63260 poke intin+4,1
63262 poke intin+6,0
63264 poke intin+8,1
63266 vdisys
63268 out 2,7
63270 return
```

Routine: MAUSDATA

```
5     ' merge "MAUSDATA.bas"
6     ' merge "SHOWMAUS.bas"
9     '
10    restore wurm   : gosub mausdata :gosub showmaus
20    warte =inp(2)
30    restore hammer  : gosub mausdata :gosub showmaus
40    warte =inp(2)
50    restore kaffee  : gosub mausdata :gosub showmaus
60    warte =inp(2)
70    end
```

Listing 2: MAUS SYMBOL

Mauseditor

Das Programm 'MAUSEDITOR' dient zum einfachen Erstellen von Maussymbolen. Es rechnet die benötigten dualen Bitmuster-Werte in Dezimalwerte um und macht die sonst doch sehr mühselige Umrechnung mit Bleistift und Taschenrechner überflüssig. Weiterhin speichert sie die errechneten Werte in die zum GEM-Aufruf notwendigen Speicherzellen.

Nach Eingabe von

Gosub MAUSEDITOR

erscheint das neu berechnete Maussymbol auf dem Bildschirm.

Die dezimalen Werte des Mausmusters und seines Hintergrundmusters sollten nach Fertigstellung notiert und aus Geschwindigkeitsgründen in Data-Zeilen abgelegt werden (siehe Programm Mausdata).

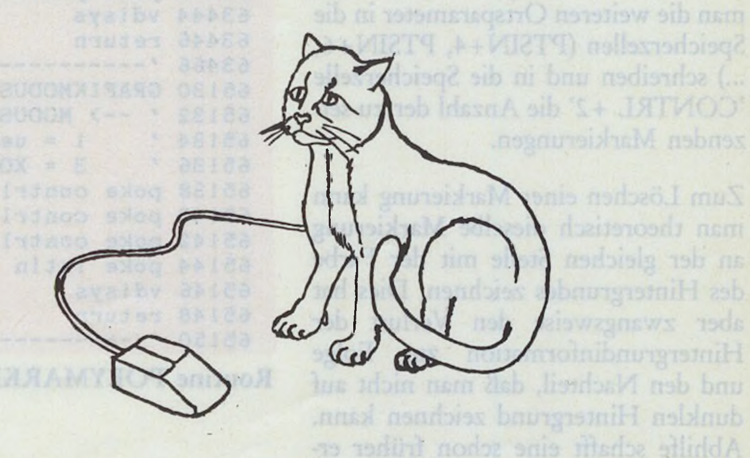
Bei jedem Diskettenzugriff schaltet der Basic-Interpreter wieder den normalen Mauspfeil ein. Das neudefinierte Symbol kann in diesem Falle wieder eingeschaltet werden durch

Gosub Mausneu

Falls Sie durch dieses Programm angeregt werden und selbst eigene Maussymbole entwerfen, so können Sie uns diese gerne zuschicken. Wir werden einige in den nächsten Ausgaben veröffentlichen.

```
63050 MAUSEDITOR:
63052 dim a$(16),b$(16)
63054 a$( 0)="---*--*----*----":b$( 0)="--******--***---"
63056 a$( 1)="--*--*----*-----":b$( 1)="-******--****---"
63058 a$( 2)="---*--*---*-----":b$( 2)="--**********----"
63060 a$( 3)="---*--*--*------":b$( 3)="--*********-----"
63062 a$( 4)="---*---*--*-----":b$( 4)="--*********-----"
63064 a$( 5)="----*-*--*------":b$( 5)="---*******------"
63066 a$( 6)="----------------":b$( 6)="--********------"
63068 a$( 7)="--*********-----":b$( 7)="-*************--"
63070 a$( 8)="--************--":b$( 8)="-**************-"
63072 a$( 9)="--*********--**-":b$( 9)="-***************"
63074 a$(10)="--*********---*-":b$(10)="-***************"
63076 a$(11)="--*********--**-":b$(11)="-***************"
63078 a$(12)="--************--":b$(12)="-***************"
63080 a$(13)="--*********-----":b$(13)="-**************-"
63082 a$(14)="---*******------":b$(14)="--*********-----"
63084 a$(15)="----------------":b$(15)="---*******------"
63086 for a=0 to 15
63088 mustervor =0 :musterhinter =0
63090 for b=15 to 0 step -1
63092 if mid$(a$(a),b+1,1)="*" then bit=1 else bit=0
63094 mustervor = mustervor+2^(15-b)*bit
63096 if mid$(b$(a),b+1,1)="*" then bit =1 else bit =0
63098 musterhinter=musterhinter+2^(15-b)*bit
63100 next
63102 poke intin +a*2+42 ,mustervor
63104 poke intin +a*2+10 ,musterhinter
63106 ' lprint mustervor,musterhinter
63108 next
63110 goto mausneu
63112 '-----------------------------------------------
63250 mausneu:
63252 poke contrl,111
63254 poke contrl+6,37
63256 poke intin,5
63258 poke intin+2,5
63260 poke intin+4,1
63262 poke intin+6,0
63264 poke intin+8,1
63266 vdisys
63268 out 2,7
63270 return
63272 '---------------
```

Routine MAUSEDITOR



Polymarker

Es handelt sich hierbei um verschiedene Zeichen, die als Markierungen auf den Bildschirm gesetzt werden können. Es stehen folgende Zeichen (siehe Bild) zur Verfügung:

- Punkt
- Kreuz
- Stern
- Rechteck
- schräges Kreuz
- Raute

Außer dem Punkt können alle diese Markierungsarten in nahezu beliebiger Größe dargestellt werden.

Der Aufruf erfolgt durch:

XPOS = .. : YPOS = .. :
MARKERTYP = .. :
MARKERGROESSE = .. :
GOSUB POLYMARKER

Zum Darstellen einer weiteren Markierung kann man das Unterprogramm aufrufen mit:

XPOS = .. : YPOS = .. :
GOSUB MARKERSETZEN

Falls man gleichzeitig mehrere Markierungen setzen will, beispielsweise zum Markieren bestimmter Punkte in einem Diagramm, so ist dies auch mit dieser Routine möglich. Dazu muß man die weiteren Ortsparameter in die Speicherzellen (PTSIN+4, PTSIN+6, ...) schreiben und in die Speicherzelle 'CONTRL +2' die Anzahl der zu setzenden Markierungen.

Zum Löschen einer Markierung kann man theoretisch dieselbe Markierung an der gleichen Stelle mit der Farbe des Hintergrundes zeichnen. Dies hat aber zwangsweise den Verlust der Hintergrundinformation zur Folge und den Nachteil, daß man nicht auf dunklen Hintergrund zeichnen kann. Abhilfe schafft eine schon früher erwähnte GEM-Routine ('Grafikmodus'),

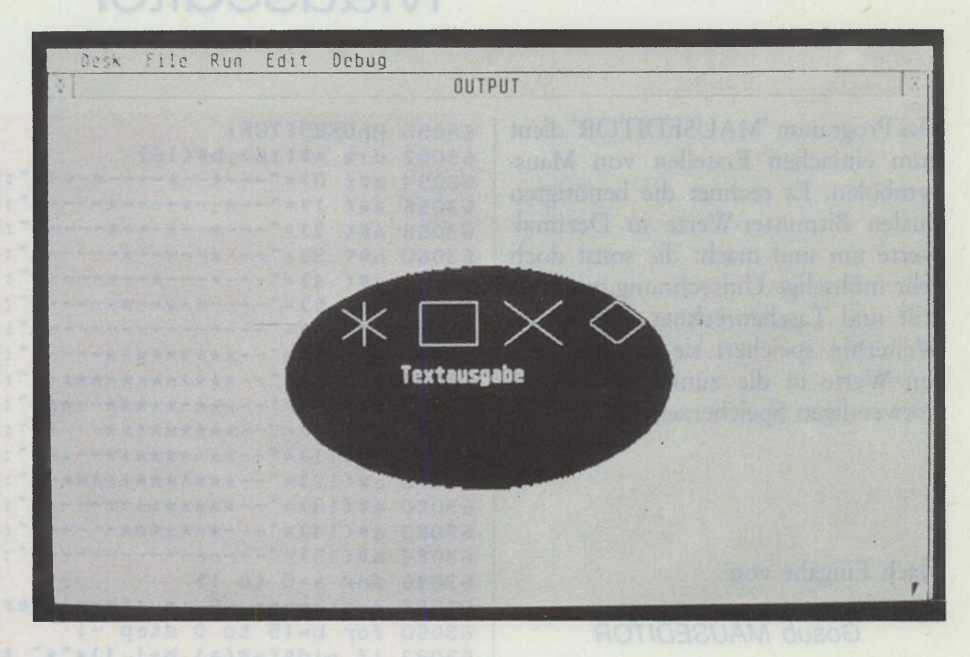


Bild 2: Polymarker & Grafikmodus 3

```
63400 polymarker:
63402 '--> XPOS ; YPOS
63404 '--> MARKERTYP ; MARKERGROESSE
63406 '
63408 poke contrl,18
63410 poke contrl+2,0
63412 poke contrl+6,1
63414 poke intin, markertyp
63416 vdisys
63418 poke contrl,19
63420 poke contrl+2,1
63422 poke contrl+6,0
63424 poke ptsin,0
63426 poke ptsin+2 ,markergroesse
63428 vdisys
63430 '
63432 markersetzen:
63434 ' (--> XPOS ; YPOS)
63436 poke contrl,7
63438 poke contrl+2,1: ' oder mehr
63440 poke ptsin ,xpos+1
63442 poke ptsin+2  ,ypos+38
63444 vdisys
63446 return
63466 '--------------------
65130 GRAFIKMODUS:
65132 ' --> MODUS
65134 '    1 = ueberschreiben        2 = mischen
65136 '    3 = XOR-verknuepfen       4 = revers,mischen
65138 poke contrl ,32
65140 poke contrl+2,0
65142 poke contrl+6,1
65144 poke intin ,modus
65146 vdisys
65148 return
65150 '--------------------------------------------
```

Routine POLYMARKER

welche den Modus bestimmt, mit der Text oder Grafik auf dem Bildschirm erscheinen. Für unsere Anwendung ist Modus '3' besonders geeignet. Es handelt sich hierbei um eine Exclusiv-ODER Verknüpfung des zu setzenden Bildpunktes mit dem schon vorhandenen. Durch zweimaliges Aufrufen einer beliebigen Bildschirmausgabefunktion, erhält man wieder die ursprüngliche Bildschirminformation; der Hintergrund bleibt also erhalten. Interessant ist dies beim kurzzeitigen Anzeigen von Markierungen, oder wenn man z. B. einen Kommentar in das Bild einblenden will, der später wieder verschwinden soll, ohne den Hintergrund mit zu löschen (siehe Bild 2).

```
5     ' merge "POLYMARKER.bas"
6     ' merge "GRAFIKMODUS.bas"
9     '
10    fullw 2: clearw 2
20    color 1,1,1,1,1
30    modus = 3: gosub grafikmodus
40    pellipse 310,200,140,80 ,0,3600
50    start:
60    xpos=50 : ypos=160:markergroesse=50
70    for markertyp = 1 to 7
80    xpos=xpos+60
90    gosub polymarker
100   next
110   gotoxy  15,11   : print "Textausgabe"
120   warte = inp(2)
130   goto start
```

Listing 3: Markierungen

Polyline

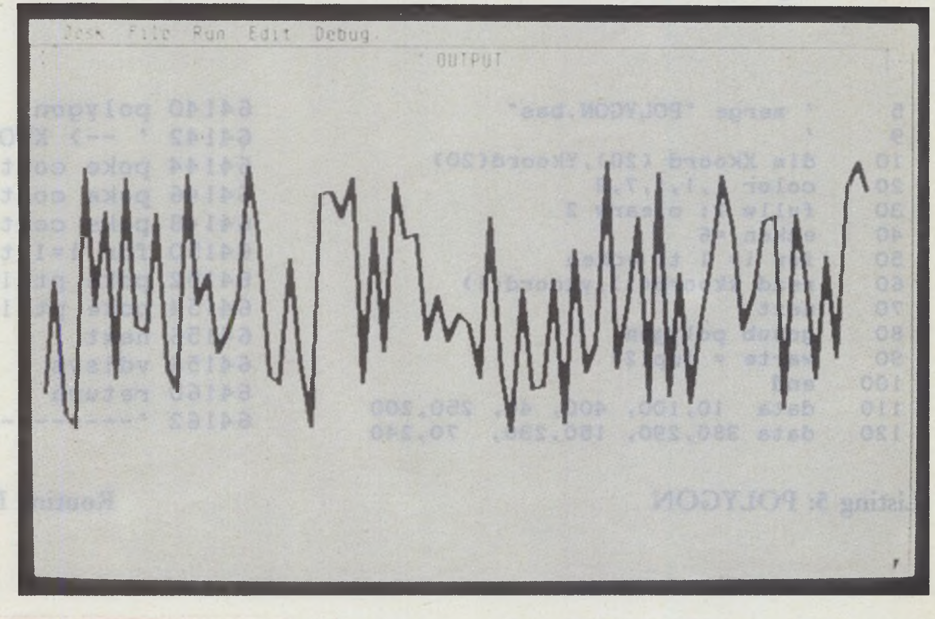
Mit Hilfe der 'POLYLINE'-Funktion können mehrere Punkte verbunden werden. Prinzipiell ist dies auch mit dem normalen 'LINEF'-Befehl möglich, aber die GEM-Routine arbeitet schneller (siehe Beispiel). Diese Routine eignet sich beispielsweise zum Verbinden von errechneten Werten (in einem Diagramm) oder sonstiger Punkte (siehe Bild 3). Ein weiteres Anwendungsgebiet wäre das Verknüpfen von Eckpunkten eines dreidimensionalen Körpers, da es dort besonders auf hohe Geschwindigkeit ankommt.

Aufruf durch

```
XKOORD ( 1,...,n)
YKOORD ( 1,...,n)
Gosub POLYLINE
```

```
64300 POLYLINE:
64302 poke contrl,6
64304 poke contrl+6,0
64306 poke contrl+2,anzahl
64308 for i=0 to anzahl
64310 poke ptsin +i*4,xkoord(i)+1
64312 poke ptsin +2+i*4,ykoord(i)+38
64314 next
64316 vdisys
64318 return
64320 '--------------------
```

Routine POLYLINE




```
5     'merge "polyline.bas"
9     '
10    dim xkoord(200),ykoord(200)
20    fullw 2:clearw 2
30    anzahl=80
40    for i=0  to anzahl
50    xkoord(i) = 10+i*600/anzahl
60    ykoord(i) = 100+rnd(1)*200
70    next
80    gosub polyline
90    warte=inp(2)
100   end
```

Listing 4: Linienbefehl

Füllfläche

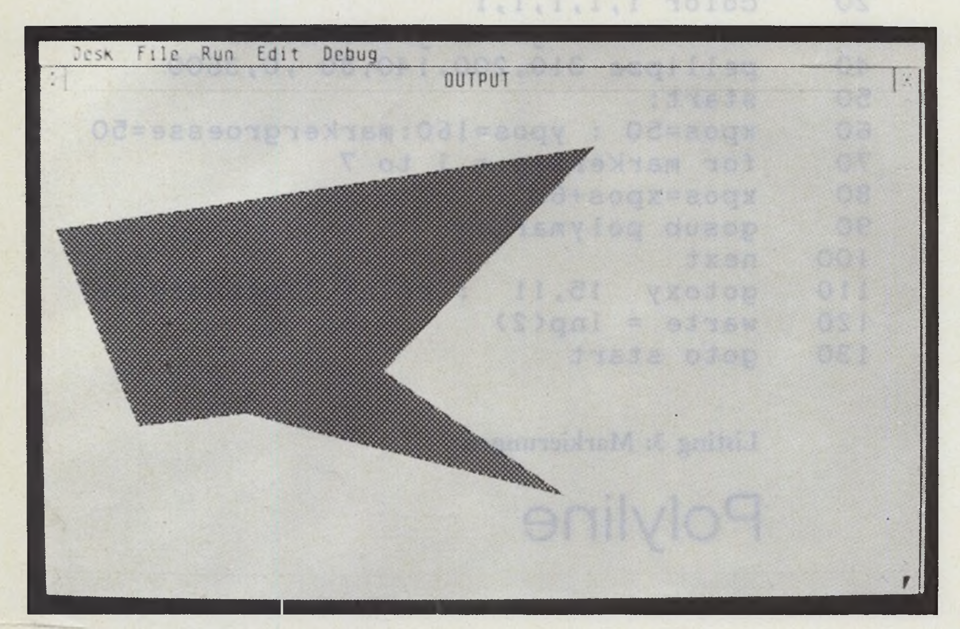


Bild 4: ausgefüllte Fläche

Ein ähnlicher Befehl wie der vorherige, aber weitaus interessanter, ist der 'POLYGON'-Befehl. Mit ihm können Flächen zwischen verschiedenen Punkten ausgemalt werden (Bild 4). Im Gegensatz zu einem Basic-Programm mit 'LINEF' und 'FILL'-Befehlen arbeitet diese GEM-Routine wiederum wesentlich schneller. Es können hiermit auch alle erdenklichen geometrische Gebilde erzeugt werden. Das Füllmuster kann mit dem 'COLOR'-Befehl definiert werden.

Aufruf durch

```
XKOORD ( 1,...,n)
YKOORD ( 1,...,n)
Gosub POLYGON
```

```
5     ' merge "POLYGON.bas"
9     '
10    dim Xkoord (20),Ykoord(20)
20    color 1,1,1,7,2
30    fullw 2: clearw 2
40    ecken =6
50    for i= 1 to ecken
60    read xkoord(i),ykoord(i)
70    next
80    gosub polygon
90    warte = inp(2)
100   end
110   data  10,100, 400, 40, 250,200
120   data 380,290, 150,230,  70,240
```

Listing 5: POLYGON

```
64140 polygon:
64142 ' --> KOORD ( )
64144 poke contrl,9
64146 poke contrl+6,0
64148 poke contrl+2,ecken
64150 for i=1 to ecken
64152 poke ptsin+(i-1)*4,xkoord( )+1
64154 poke ptsin+2+(i-1)*4,ykoord(i)+38
64156 next
64158 vdisys
64160 return
64162 '---------------------
```

Routine POLYGON

Vieleck

Dabei handelt es sich um eine spezielle Anwendung des POLYGON-Befehls. Er bietet die Möglichkeit, symmetrische Vielecke mit beliebiger Eckenzahl zu erzeugen (siehe Bild 5). Dazu gehören beispielsweise Dreiecke oder Vierecke, deren Ausrichtung zur horizontalen Achse beliebig ist. Wünscht man eine andere Lage der Figuren, so kann man den Anfangs- und Endwinkel im Unterprogramm entsprechend verschieben.

Der Aufruf erfolgt durch:

XPOS= .. : YPOS = :
RADIUS= .. : ECKEN = .. :
GOSUB VIELECK

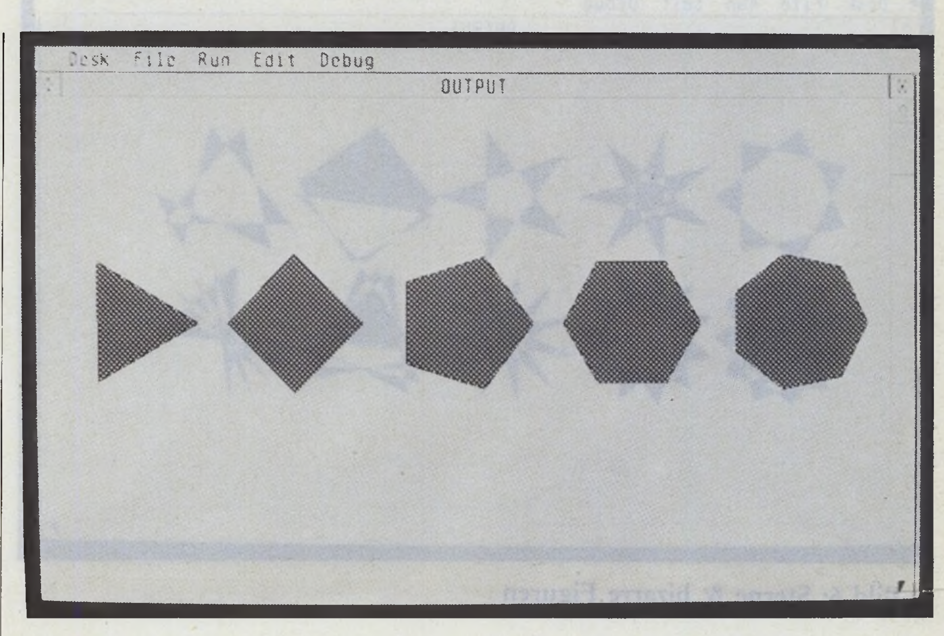


Bild 5

Das Füllmuster des entstehenden Vieleckes sollte vor dem Aufruf mit dem 'COLOR'-Befehl definiert werden.

```
5       ' merge "VIELECK.bas"
9       '
10      fullw 2:clearw 2
20      color 1,1,1,7,2
30      xpos=-50 : ypos =150
35      radius=50
40      for ecken = 3 to 7
50      xpos =xpos+120
55      gosub vieleck
60      next
70      warte = inp(2)
80      end
```

Listing 6: Vieleck

```
64100 VIELECK:
64102 ' --> XPOS ; YPOS
64104 ' --> RADIUS ; ECKEN
64106 phi=3.141593/ecken/2
64108 stern:
64110 ' --> PHI
64112 poke contrl,9
64114 poke contrl+6,0
64116 poke contrl+2,ecken
64118 for eck=0 to ecken*4 step 4
64120 poke ptsin +eck    , 1+xpos+cos(phi*eck)*radius
64122 poke ptsin+(eck+2),38+ypos+sin(phi*eck)*radius
64124 next
64126 vdisys
64128 return
64130 '-----------------------------------------
```

Routine VIELECK

Stern

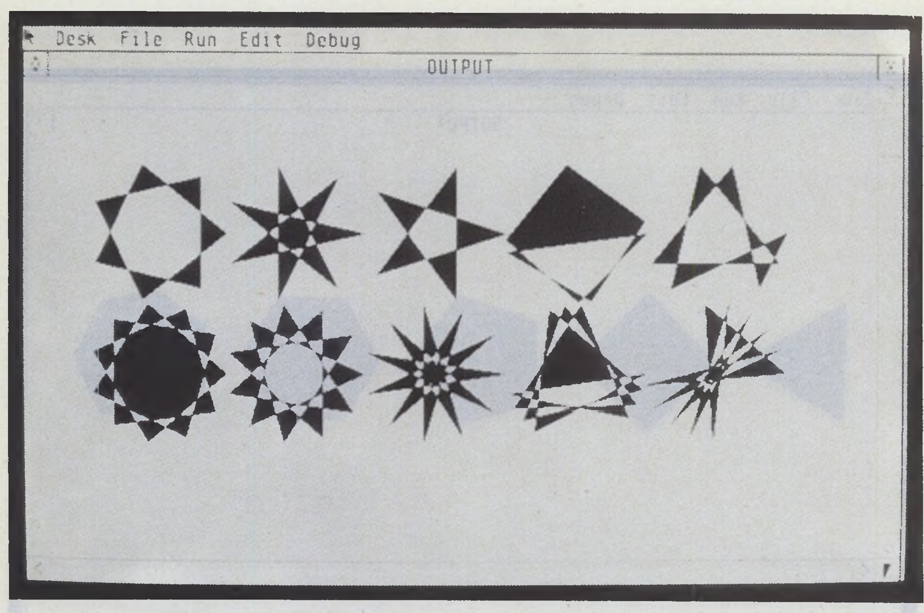


Bild 6: Sterne & bizarre Figuren

Obwohl es, unter Berücksichtigung der Jahreszeit, eher angebracht wäre, Ostereier zu produzieren, so soll doch auf diesen Weihnachtsnachtrag eingegangen werden. Es handelt sich hierbei um die leicht abgeänderte Form der vorherigen 'Vieleck-Routine', die, in Abhängigkeit von dem Winkel 'Phi', teils Sterne und teils bizarre Muster produziert (Bild 6). Beachtenswert ist hierbei die Geschwindigkeit, mit der eine solche Figur gezeichnet wird.

(HS)

Fortsetzung folgt:

- Farbbefehle
- AES-Routinen

```
5      ' merge "VIELECK.bas"
9      '
10     fullw 2:clearw 2
20     color 1,1,1,8,2
30     ypos =150 :radius=50
40     ecken =7
50     xpos =100:phi=3.1416/2/ecken*2   : gosub stern
60     xpos =200:phi=3.1416/2/ecken*3   : gosub stern
70     xpos =300:phi=3.1416/2/ecken*2.8: gosub stern
80     xpos =400:phi=3.1416/2/ecken*1.7: gosub stern
90     xpos =500:phi=3.1416/2/ecken*2.5: gosub stern
100    ecken =11:ypos=250
110    xpos =100:phi=3.1416/ecken*1.5   : gosub stern
120    xpos =200:phi=3.1416/ecken*2     : gosub stern
130    xpos =300:phi=3.1416/ecken*2.5   : gosub stern
140    xpos =400:phi=3.1416/ecken*1.9   : gosub stern
150    xpos =500:phi=3.1416/ecken*2.6   : gosub stern
160    warte=inp(2)
170    end
```

Listing 7: Sterne

WordStar
auch für den ATARI ST

Mit dem CP/M 80 Emulator hat sich dem ATARI ST kurzfristig eine große Softwarequelle eröffnet. Zwar brennt der ATARI im Z80 Modus nur auf Sparflamme aber ein reichhaltiges Programmangebot dürfte dies vorerst vergessen machen. Ein wichtiges Beispiel dieser Softwarepalette ist Wordstar.

WordStar ist, wie der Name schon sagt, ein Textverarbeitungsprogramm. Allerdings grenzt die Aussage „ein Textverarbeitungsprogramm" schon an eine Untertreibung. Denn, WordStar ist das am meisten verbreitete Textverarbeitungssystem, das für Personal Computer entwickelt wurde. Viele andere Textverarbeitungsprogramme basieren auf dem Konzept von WordStar, somit kann man sagen, daß wenn man mit WordStar umzugehen versteht, man mit fas jedem anderen Textverarbeitungsprogramm ohne große Einarbeitung zurechtkommt.

Wozu brauche ich ein Textverarbeitungsprogramm, ich kann meine Briefe auch wie bisher auf einer Schreibmaschine schreiben? Zu dieser Frage eines voraus: Jemand, der einmal einen Brief auf einem Textverarbeitungssystem (und einen Personal Computer mit Textverarbeitungsprogramm und Drucker kann man als solches bezeichnen) geschrieben hat, möchte die Annehmlichkeiten, die einem geboten werden, nicht mehr missen.

Wer hat sich nicht schon beim Schreiben eines Textes auf einer Schreibmaschine über den mit Abnehmen der Zehn-Finger-Fertigkeit anwachsenden Verbrauch von Korrekturhilfen geärgert, und wer hätte nicht gerne nach Vollendung des Textes noch hier und da ein paar Leerzeilen eingefügt, und wer hat nicht schon erst im Nachhinein festgestellt, daß der von ihm in Mühe erstellte Text auf einmal voller Fehler steckt?

```
      A:EINLADUN  PAGE 1 LINE 10 COL 50          INSERT ON
                 < < <      M A I N   M E N U     > > >
   --Cursor Movement--   ! -Delete- !  -Miscellaneous-  !  -Other  Menus-
 ^S char left ^D char right !^G  char  ! ^I Tab   ^B Reform ! (from Main only)
 ^A word left ^F word right !DEL chr lf! ^V INSERT ON/OFF   !^J Help  ^K Block
 ^E line  up  ^X line down  !^T word rt!^L Find/Replce again!^Q Quick ^P Print
     --Scrolling--          !^Y  line  !RETURN End paragraph!^O Onscreen
 ^Z line up   ^W line down  !          ! ^N Insert a RETURN !
 ^C screen up ^R screen down!          ! ^U Stop a command  !
L----!----!----!----!----!----!----!----!----!----!----!---------R


          Hallo Ines

Wenn Du am 24.Februar ein wenig Zeit hast, kannst Du zum grossen
PIZZA-Essen bei mir vorbeischauen.

                                        Dein Theo

PS:  Aber bitte nicht erst um Mitternacht - Danke !!
```

Wäre dieser Text doch nur mit Hilfe eines Textsystemes geschrieben worden, so bräuchte man sich über solche Probleme keinerlei Gedanken machen, denn ein Textverarbeitungsprogramm nimmt einem die Lösung solcher Probleme hervorragend ab.

Damit Sie einmal eine Vorstellung bekommen, wie so etwas auf dem Bildschirm aussieht: Der geschriebene Text ist auf dem Bildschirm zu sehen und zwar in der Anordnung wie man ihn eingegeben hat. Damit nicht genug, denn der Text wird nach Fertigstellung auch so ausgedruckt, wie man ihn auf dem Bildschirm sieht.

Wäre das der ganze Vorteil, den WordStar bietet, so wäre das nicht gerade viel und die Anschaffung hierfür würde sich nicht gerade lohnen. Aber nein, mit WordStar, das von MicroPro International Corp. in den USA entwickelt wurde, können Sie

- Ihre Text formatiert (d. h. Ränder und Tabulatorenstellen nach eigenen Wünschen gesetzt) eingeben
- den einmal eingegebenen Text nach und während der Eingabe in Bezug auf rechten und linken Rand ändern und umformatieren, wobei WordStar automatisch Trennhilfen erteilt, um Worte nicht grammatikalisch falsch automatisch zu trennen
- im Text seitenweise auf dem Bildschirm vor- und zurückblättern
- einzelne Buchstaben, Worte, Zeilen und Absätze löschen und nachträglich wieder einzelne Buchstaben, Worte, Zeilen und Absätze, oder auch schon einmal geschriebene Texte in den gerade bearbeiteten Text einfügen
- den Text rechtsbündig (wie in einer Zeitungsspalte) schreiben
- Worte und Zeilen zentrieren
- einzelne Buchstaben oder Worte oder auch mehr im ganzen Text, oder in einem Teil des Textes automatisch suchen und gegebenenfalls tauschen

- bestimmte Textsequenzen als Blöcke definieren und diese dann in dem gerade in Bearbeitung befindlichen Text an irgend eine Stelle kopieren oder verschieben oder löschen oder in eine andere Datei schreiben, d.h. der als Block definierte Text wird auf die Diskette kopiert
- immer wiederkehrende Absätze und Formulierungen auf der Diskette speichern und zu einem neuen Text je nach Bedarf zusammensetzen
- Text unterstreichen, durch Doppelt- oder Dreifachanschlag beim Ausdruck hervorheben, Text hoch- oder tiefstellen
- auch jeden Drucker mit den verschiedensten Typenrädern ansprechen.

Dies sind nur die wichtigsten Funktionen, die WordStar bietet. Damit WordStar diese ganzen Funktionen ausführt, ist natürlich ein bestimmter Befehlssatz notwendig. WordStar lebt quasi von der CONTROL-Taste, denn sei es, man möchte einen Buchstaben löschen, oder Blöcke definieren, oder auch nur den geschriebenen Text auf der Diskette abspeichern, es ist auf alle Fälle die CONTROL-Taste, gedrückt zusammen mit einer anderen Taste, notwendig. Diese Tatsache erschwert es dem Anfänger erheblich, seinen allerersten Brief zustande zu bringen. Nachdem man das Programm WordStar geladen hat ist zwar im allgemeinen beim Start des Programms automatisch eine „Hilfsfunktion" eingeschaltet, welche es ermöglicht, die gängigsten Befehle auf dem Bildschirm über dem zu bearbeitenden Text wiederzufinden, aber es sei an dieser Stelle gesagt, daß es dem Neuling auf diesem Gebiet am besten weiterhilft, wenn er sich ein Buch über die Handhabung von WordStar aneignet. Nur so kann man die Fülle an Vorzügen, die dieses Programm bietet, sinnvoll anwenden, und sich somit viel Arbeit bei der Erstellung eines Textes ersparen.

Das Textverarbeitungsprogramm WordStar bietet allerdings noch weitere Vorteile.

Möchte man z. B. ein Programm in der Programmiersprache „C" schreiben, und soll dies möglichst komfortabel vonsdtatten gehen, so hat man die Möglichkeit, das Programm in dem speziellen WordStar-Modus „N = Eröffnen einer Programmdatei" zu schreiben und unter den Vorzügen eines ausgereiften Texteditors unter Verwendung aller WordStar-Befehle zu editieren, um das Programm sodann von einem C-Compiler compilieren zu lassen.

Dieser spezielle „N-Modus" hat keine variable linke- und rechte Randbegrenzung wie der normale Textverarbeitungsmodus, denn er erzeugt eine reine ASCII-Datei ohne Steuerzeichen. Dieser Editor wird den Benutzern von TURBO-Pascal schon ein Begriff sein, denn TURBO-Pascal arbeitet mit dem selben Editor.

Zu einer ganz erheblichen Arbeitserleichterung kommen wir nun: Man stelle sich einmal vor, man möchte an ca. 50 verschiedene Personen eine Einladung zu einer Party verschicken, wobei der Text der Einladung immer der gleiche ist, es ändern sich lediglich die Anreden und die Adressen von Brief zu Brief

Was macht man in so einer Situation?

Lösung 1:
Man setzt sich hin und tippt jeden Brief individuell neu

Lösung 2:
Man läßt jeden Brief individuell neu tippen

Lösung 3:
Man zählt sich zu den glücklichen Besitzern des zu WordStar kompatiblen Softwarepaketes MailMerge, schreibt den Brief nur einmal und läßt im Mischdruck-Verfahren die verschiedenen Anreden und Adressen an die richtigen Stellen in der Einladung automatisch einfügen, um sodann auch noch die Adressaufkleber für die Briefumschläge drucken zu lassen.

Das Programm MailMerge ist nämlich, wie oben schon erwähnt, ein unabhängiges, jedoch mit WordStar voll kompatibles Softwarepaket, das es erlaubt, einen allgemein gehaltenen Brief zu schreiben, in dem an den individuellen Textstellen Variable geschrieben werden, die dann im Mischdruck-Verfahren mit den individuellen Daten aus einer anderen Datei, oder auch während des Druckes von der Tastatur aus, ausgetauscht werden.

Stehen die einzufügenden Daten in einer dBase II (III)-Datei zur Verfügung, so kann auch auf diese Daten ohne weiteres zugegriffen werden.

Nun noch zu einem weiteren Sopftwarepaket, das in engem Zusammenhang mit WordStar steht, nämlich eines, welches jene unter uns ansprechen wird, die nach vollbrachter schreiberischer Leistung immer noch gänzlich im Dunkel tappen, ob denn das Geschriebene auch der deutschen Rechtschreibung entspricht, oder nicht.

Es handelt sich um das Programm SpellStar, welches übrigens genau wie Mailmerge auch von WordStar aus aufgerufen werden kann.

SpellStar enthält ein „Wörterbuch" von 20 000 Worten, kann jedoch noch erweitert werden, damit auch Spezialausdrücke überprüft werden können.

Möchte man näheres über die hier genannten Programme erfahren, so lohnt es sich auf jeden Fall ein Buch über dieses Theme zu lesen, denn nur so kann man sich viel unliebsame Arbeit ersparen und auch noch die Vorzüge von MailMerge und SpellStar beim Schreiben eines Textes genießen.

B W



Ein Datenbanksystem mit der Maus

Datenbanken gehören zu den wichtigen Anwendungsbereichen in der Computerwelt. Ein Rechner, für den kein solches Programm existiert, wird nicht als professionell betrachtet.

Datenbankenprogramme helfen bei der Erfassung großer Mengen an Daten, wie das z. B. bei einem Büro der Fall ist, wobei die Personaldaten in der Regel in einer Kartei gesammelt werden. Ein Datenbankprogramm ist ein elektronischer Ersatz solcher Register.

Je besser ein Dateiprogramm ist, desto reicher die Möglichkeiten, die es anbietet. Insofern gibt es Programme, die die Durchführung einfacher, tabellarischer Rechnungen sowie Einfügen von kleinen Texten ermöglichen.

Es gibt in dem PC-Bereich seit langem Programme dieser Art, die sich durch ihre leistungsfähigen Merkmale einen Platz auf dem Markt gesichert haben. Eines davon ist der weltbekannte D-BASE von Micro-Soft. D-BASE ist mehr als ein Dateiverwaltungsprogramm, es ist eine Sprache, um Datenbanken zu erzeugen. Ein effektiver Einsatz von D-BASE ist nur nach einem intensiven Lernprozess dieses Programms möglich.

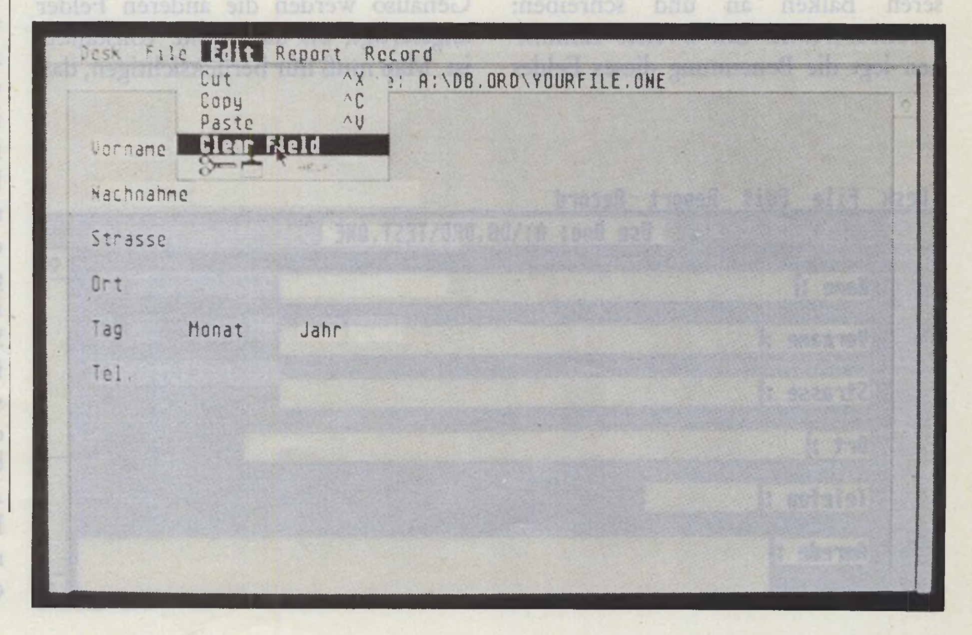
In dem Zeitalter benutzfreundlicher Oberfläche, wie z. B. GEM ein richtiger Vertreter ist, wäre es ein Anachronismus für Rechner, die solche Betriebssysteme besitzen, Dateiprogramme zu entwickeln, die die obengenannten Möglichkeiten nicht ausnutzten.

STOLLWARE bietet für die neue ST-Generation (520 ST + und 260 ST) von Atari en Produkt, das konsequent die Fähigkeiten eines solchen Betriebssystems verwendet.

Das Datenbanksystem von STOLLWARE ist in zwei Programme gegliedert: das eine ist für die Herstellung von Masken vorgesehen, das andere ist das Dateiverwaltungsprogramm an sich.

Am Anfang war die Maske...

Durch Anklinken des Programms MAKE ONE landet man in dem schon bekannten Bild, das alle Programme, die unter GEM arbeiten, kennzeichnet; Oben steht die MENÜ-



BAR, gleich darunter die Arbeitsfläche.

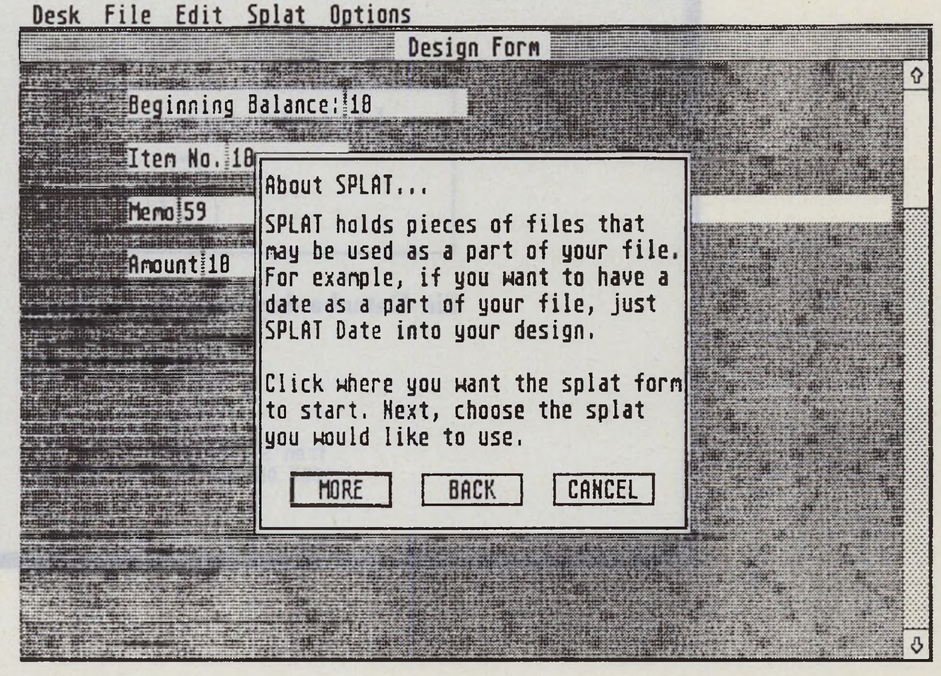
Das DESK-MENÜ muß mittlerweise jedem ST-Anwender bekannt sein, und wir brauchen dies hier nicht erneut beschreiben.

Um eine Maske herzustellen, muß man zuerst bei dem FILE-MENÜ die Auswahl NEW anklinken. Jetzt steht die gesamte Arbeitsfläche für die Editierung zur Verfügung. Die MAUS wird nun verlangt. Der größte Teil der Editierung wird durch Platzierung der Pfeilsymbole und durch zusätzliches Anklinken mit der MAUS verrichtet.

Eine Maske besteht in der Regel aus vielen verschiedenen Feldern. Diese Felder werden mit einem Namen versehen; direkt dahinter wird ein Platz für den Inhalt (Eintragung) reserviert. Bei DB-MASTER ONE müssen Sie mit dem Pfeil jedes einzelne Feld auf dem Bildschirm positionieren. Nach Betätigung der linken Taste auf der MAUS erscheint ein leerer Balken mit dem augenblicklich freien Platz (was später beliebig verändert werden kann). Dieser Balken entspricht unserem Eintragungsfeld, das jetzt gekennzeichnet werden muß.

Am Beispiel einer einfachen Adress-Datei werden wir die Herstellung einer Maske verdeutlichen. Unser erstes Feld trägt den Namen NACHNAME.

Wir klinken mit der Maustaste an unseren Balken an und schreiben: NACHNAME. Ein zweites Einklinken legt die Benennung dieses Feldes



fest. Rechts von dem Feld liegt der freie Platz für die Feldeintragung, deren Länge noch festgelegt werden muß. Bewegen Sie den Pfeil bis zum Ende des Balkens und klinken Sie erneut an. Jetzt können Sie die gewünschte Länge durch Bewegung der Maus beliebig ändern. Die Anzahl der Stellen wird ständig links auf dem Balken gezeichnet. Die Länge einer Eintragung beschränkt sich nicht auf eine Zeile, sondern kann mehrere umfassen. Ein so erstelltes Feld wird durch RETURN oder erneutes Anklinken quittiert.

Genauso werden die anderen Felder angefertigt, bis die Maske vollständig ist. Man muß nur berücksichtigen, daß die größtmögliche Anzahl an Feldern 100 beträgt, wobei eine Dateisatzlänge von hundert Stellen je Feld nicht überschritten werden sollte.

Zur Editierung stehen dem Anwender andere benutzfreundliche Hilfsmittel zur Verfügung. Bei dem EDIT-MENUE kann man wählen zwischen der Löschung ganzer Felder, Ersetzen von Teilen des Textes, usw. Verschiedene Schriftarten können bei dem OPTION MENÜ gewählt werden. Auch die Größe jeder einzelnen Schriftart kann stufenmäßig geändert werden. Das Programm unterscheidet zwischen dem Text des Namensfeldes und dem Text bei dem Satzfeld. Damit ist es z. B. möglich, das Namensfeld fett zu schreiben und die Eintragung kursiv, heller oder dunkler zu gestalten. Bei dem SPLAT-MENÜ handelt es sich um eine Sammlung von fertiggestellten Masken, die als Teile anderer Masken verwendet werden können. Stellen Sie sich vor, daß Sie eine Maske fertiggestellt haben und Sie möchten sie mit noch einem Feld versehen, das das Datum erfassen soll. Klinken Sie bei dem SPLAT-MENÜ die DATE-Auswahl an und jetzt können Sie das Datum-Feld platzieren. Alle diese Teilmasken können beliebig geändert werden, auf verschiedene Stellen platziert

Desk File Edit Report Record
Use One: A:\DB.ORD\TEST.ONE
Name :
Vorname :
Strasse :
Ort :
Telefon :
Anrede :

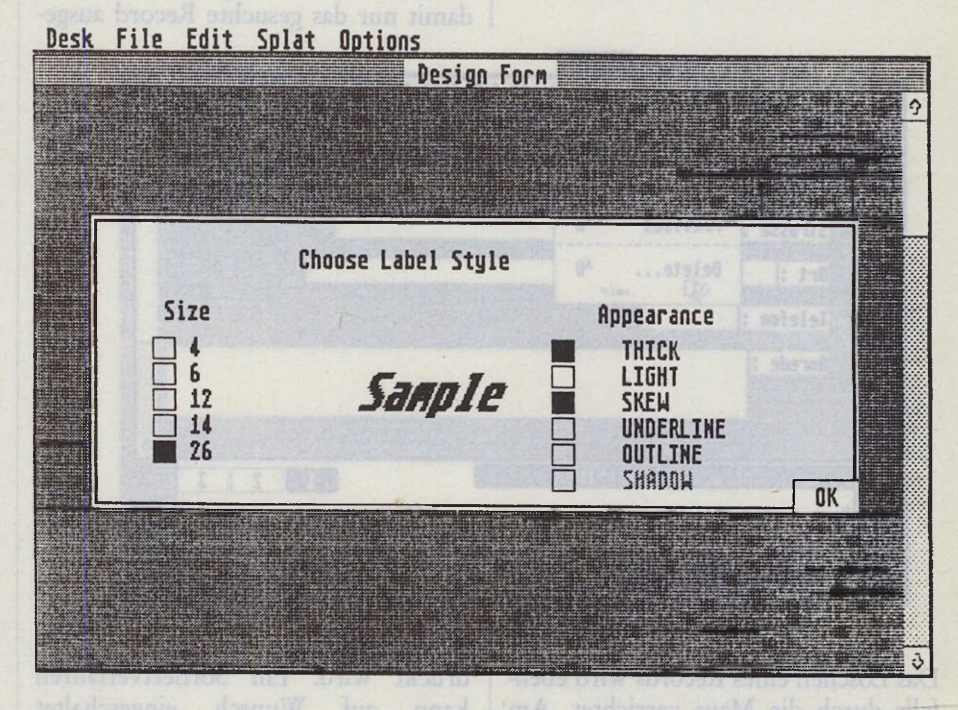
werden, können aber nicht von der Diskette gelöscht werden. Das ist insofern ein Nachteil, weil alle Feldernamen englisch geschrieben sind.

Eine fertiggestellte Maske wird durch die Auswahl SAVE bei dem FILE-MENÜ auf Diskette abgesichert. Der Name des Masken-File ist dem Benutzer frei überlassen, er muß nur die Extension ONE enthalten.

Datei-Handling

Nach der Fertigung einer Maske ist man in der Lage, eine Datei herzustellen, zu vergrößern oder zu pflegen. Dafür laden Sie das zweite Programm, das sich unter dem Namen USERONE auf der mitgelieferten Diskette befindet. Mitgeladen wird auch das Directory aller Masken-Files. Man wählt jetzt die gewünschte Maske, die auf den Bildschirm gebracht wird. Nun können die Daten eingetragen oder alte Daten (wenn vorhanden) gepflegt werden. Alle diese Ausgaben sind mit Hilfe der Maus möglich.

Um eine Eintragung zu machen, muß man das gewählte Feld zuerst aktivieren. Danach schreibt man, soweit die Länge des Feldes es erlaubt. Ein erneutes Anklinken oder die Betätigung von RETURN beendet die Eintragung in diesem Feld. Ist man bei diesem Record mit allen Eintragungen fertig, und will man mit einem neuen anfangen, muß man die Auswahl ADD bei dem RECORD-Menü oben aus dem



Menü-Ware einklinken.

Auch hier stehen dem Anwender verschiedene Hilfsmittel für die Editierung zur Verfügung. Am EDIT-Menü befinden sich mehrer Auswahlmöglichkeiten, die dem User entgegenkommen, wie z. B. das Kopieren von Texten oder Löschen des gesamten Inhalts eines Feldes usw..

Bei der Arbeit mit einer Datei spielen die Suchverfahren eine relevante Rolle. Je mehr Möglichkeiten ein Programm in diesem Bereich anbietet, desto leistungsfähiger ist es. DB-MASTER-ONE bietet vier Möglichkeiten. Drei davon sind fest eingebaut und können nur benutzt und nicht verändert werden:

SUCH ALLE (FIND ALL)
wobei alle Records angezeigt werden.

NÄCHSTE (NEXT)
Es wird das nächste Record auf dem Bildschirm gezeigt.

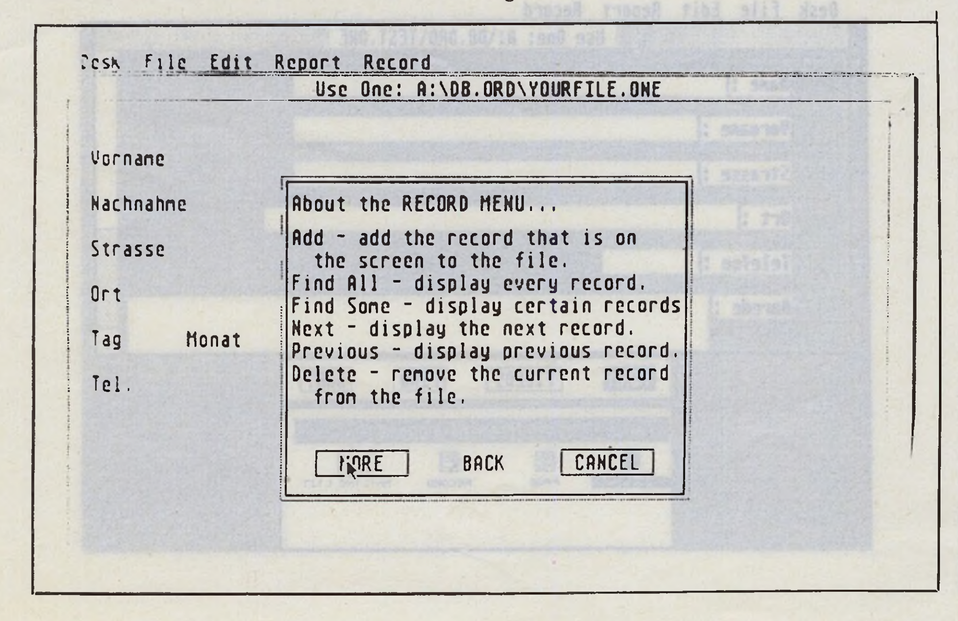
VORHER (PREVIOUS)
Das vorherige Record wird gezeigt.

Die vierte Möglichkeit steht dem Anwender frei zur Verfügung, und er kann sie – mit einigen Einschränkungen – selbst gestalten. Die Such-Schlüssel-Kriterien sind: GLEICHHEIT (EQUALS), wobei das gewünschte Feld bei einem Record mit einem gleichwertigen Feld auf Gleichheit untersucht wird.

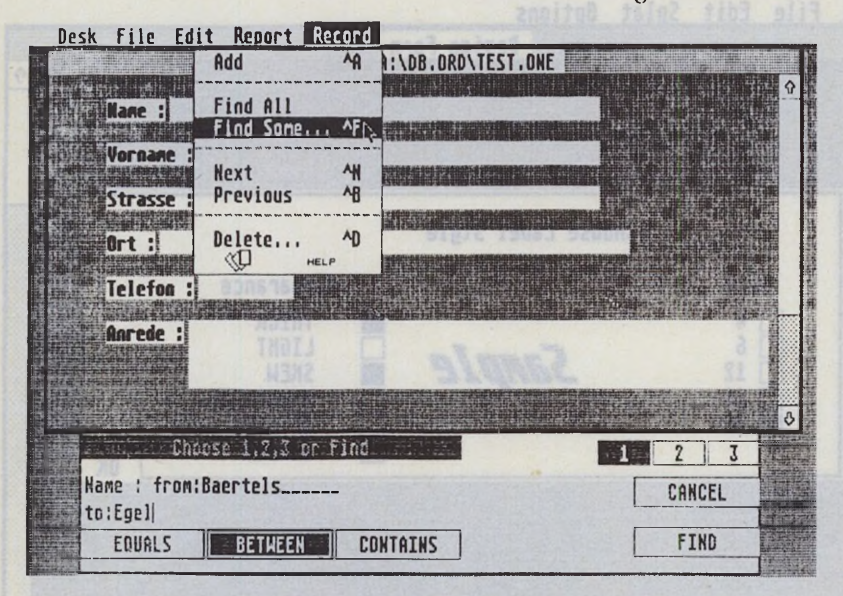
ZWISCHEN (BETWEEN)
Man kann zwischen mehreren Records (TO bis FROM) aussuchen oder zwischen mehreren gleichen Feldern.

INHALT (CONTAINS)
Die Suche wird nach dem Inhalt eines bestimmten Feldes geführt.

Jedes einzelne Sub-Kriterium kann vermischt werden. Damit vermehrt sich



die Anzahl an Möglichkeiten, die DB-MASTER-ONE zur Suche bestimmter Daten anbietet.



Das Löschen eines Records wird ebenfalls durch die Maus verrichtet. Am RECORD-Menü steht die Funktion DELETE, die das augenblickliche Record löscht. Sicherheitshalber fragt das Programm nach, ob Sie sicher sind, daß dieser Record gelöscht werden soll oder ob Sie es sich inzwischen anders überlegt haben.

Die Ausgabe erfaßter Daten kann wahlweise auf dem Bildschirm oder auf einem angeschlossenen Drucker geschehen. Da normalerweise auf dem Sichtgerät ständig die verschiedenen Ausgaben angezeigt werden, ist eine Anpassung nicht immer notwendig.

Bei DB-MASTER-ONE ist nur bei einer tabellarischen Ausgabe von mehreren Daten eine Bildschirmausgabe anpassungsbedürftig.

Anders ist es bei einer Ausgabe auf einem Drucker, wo man zuerst ein sogenanntes Protokoll erstellen muß. Das kann man nach Wunsch auf Diskette speichern, um es später immer wieder verwenden · zu können. Auch hier wird man mit mehr Auswahl konfrontiert. Die Form der Ausgabe kann in Form einer Tabelle, einer ganzen Seite oder eines einzelnes Record geschehen. Es können außerdem einzelne Felder in tabellarischer Gestalt ausgedruckt oder auf dem Bildschirm angezeigt werden, wenn man die Auswahl FIELD aktiviert. Bestimmte Suchverfahren (ähnlich wie oben beschrieben wurde) können ausgewählt werden, damit nur das gesuchte Record ausgedruckt wird. Ein Sortiertverfahren kann auf Wunsch eingeschaltet werden.

Die neuen und alten Daten werden nach einem Arbeitsgang durch Saves, aus dem FILE-MENÜ, auf Diskette abgespeichert. Alles geschieht problemlos und, was noch schöner ist, mit der Maus auch sehr bequem.

DB-MASTER-ONE ist ein ausgereiftes Programm, welches die Merkmale der schon auf dem Markt bewährten Dateienverwaltungsprogramme, längst übertrifft.

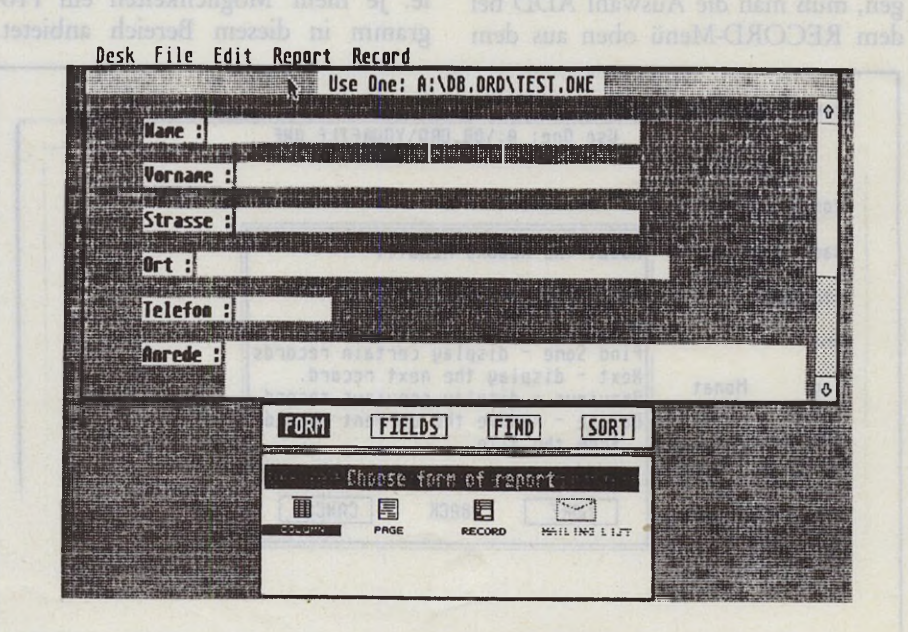
Zu bemängeln wäre, daß es sich zur Zeit um eine englische Version handelt; deswegen können z. B. keine deutshcen Laute und Meldungen englisch erscheinen. Andererseits sind die gewonnenen Daten mit anderen Programmen, wie Kalkulationstabellen oder Word-Prozessoren, nicht austauschfähig; was ein großes Manko bedeutet, vor allem wenn die gleiche Soft-Ware-Firma auch ein Textverarbeitungssystem für ST-Rechner anbietet.

Ein notwendiges Handbuch wurde auch nicht mitgeliefert, so ist der Anwender allein auf seine Eigeninitiative angewiesen. Hilfe schaffen hier nur die kurz beschriebenen Anweisungen, die das Programm selbst beinhaltet.

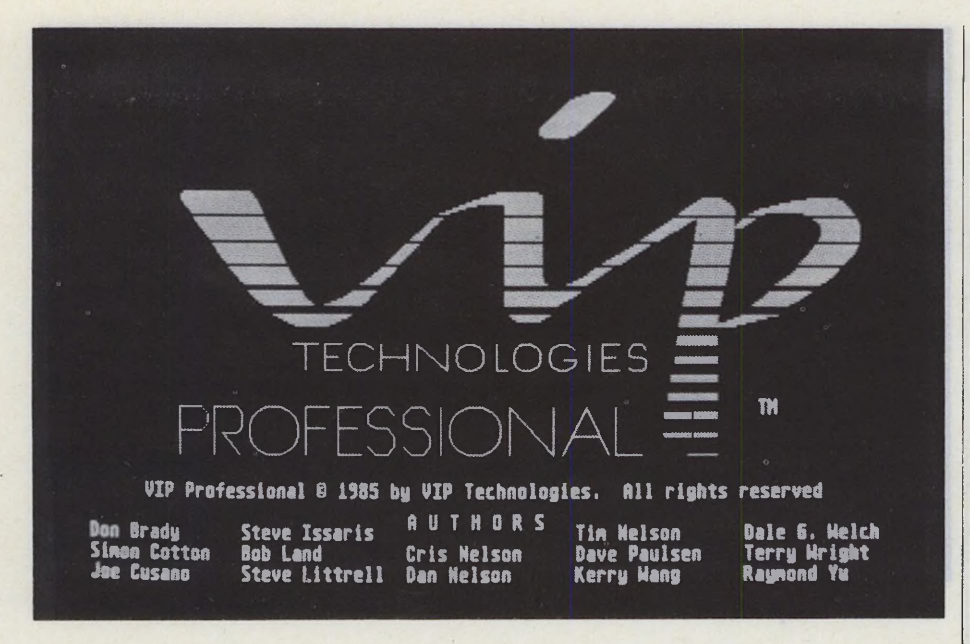
Kurz vor Redaktionsschluß wurde uns bekannt, daß Atari ein Manual (Tutorial) fertiggestellt hat und es in Zukunft mit der Diskette liefert.

Das Programm wird in Deutschland von Atari selbst vertrieben und kostet im Handel nur DM 99,–. An dieser Stelle müssen wir uns entschuldigen: Wir haben in unserer letzten Ausgabe irrtümlicherweise veröffentlicht, daß dieses Programm zum Paket des „Public-Domain"-Programm gehört und zum Nulltarif bei Atari-Händlern zu erhalten wäre. Dies war ein Mißverständnis, das wir bedauern.

(DB)



Der ST im Business eingesetzt



Mit VIP Professional bietet die Firma VIP Technologies eines der ersten professionellen Anwenderprogramme für die ATARI ST Modelle an. Es beinhaltet eine Tabellekalkulation mit Grafik und Datenbank, mit der die meisten anfallenden Büroarbeiten problemlos gelöst werden können.

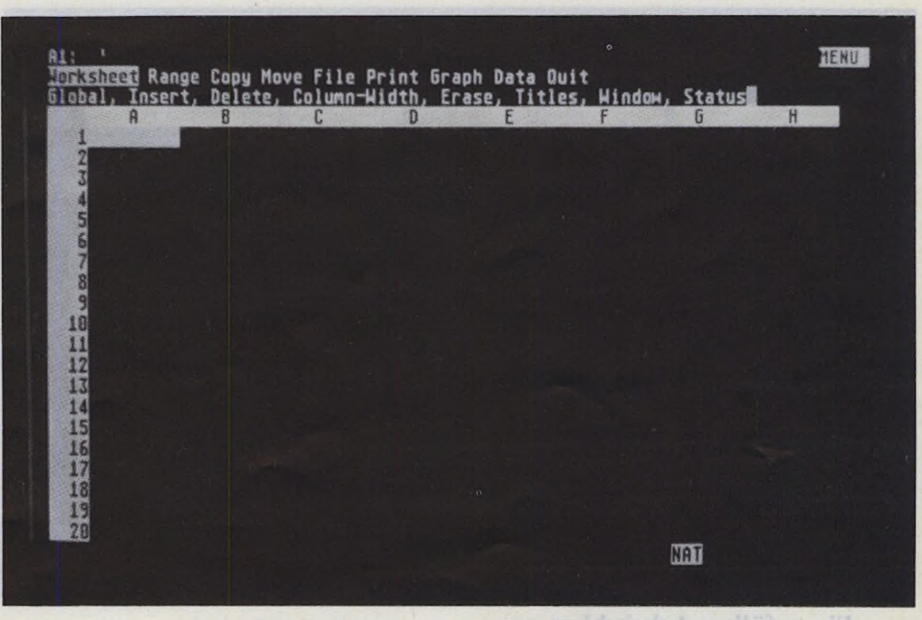
VIP ist zur Zeit nur in einer Vorabversion erhältlich, die die GEM-Möglichkeiten nicht nützt. Wenn es das TOS als ROM geben wird, soll laut Hersteller auch eine erweiterte Version auf den Markt kommen. Diese wird dann die GEM-Routinen voll einbinden, so daß die ganze Menüführung durch Pull-Down-Menüs und Maus unterstützt wird.

Auffallend an diesem Programm ist, daß es von der Bedienerführung komplett Lotus 1-2-3 für den IBM PC gleicht. Der Lieferumfang umfaßt ein ausführliches – leider noch englisches – Handbuch, eine Kurzreferenzkarte und zwei Disketten, auf denen sich das Haupt-, ein Grafikdruckprogramm und eine TOS-Version befinden. Nach dem Anklicken des Hauptprogramms kommt man nach dem Erscheinen des Firmen-Logos automatisch an das Arbeitsblatt – auch Worksheet genannt. VIP Professional arbeitet sowohl mit dem monochromen als auch dem Farbmonitor zusammen. Da es aber auf die Druck- und Farbparameter des Desktops zurückgreift, kann es nötig sein, das Programm beim ersten Mal noch einmal zu verlassen. Die Parameter können dann geändert und auf der mitgelieferten Systemdiskette abgespeichert werden. Außerdem muß vor dem Start das Datum im Desktop auf den aktuellen Stand gebracht werden, da dies im Programm nicht möglich ist. Man kann in Bezug auf diesen Punkt hoffen, daß irgendein Hardware-Entwickler eine Batteriegepufferte Echtzeituhr auf den Markt bringt, die diese Arbeit abnimmt.

Ist dann wieder das Hauptprogramm geladen, betätigt man einfach gleichzeitig die Tasten SHIFT und 7, um in das Hauptmenü zu kommen. Mit der Help-Taste oder der Funktionstaste F1 kann jederzeit ein Hilfsmenü aufgerufen werden, in dem jeder Befehl kurz beschrieben ist.

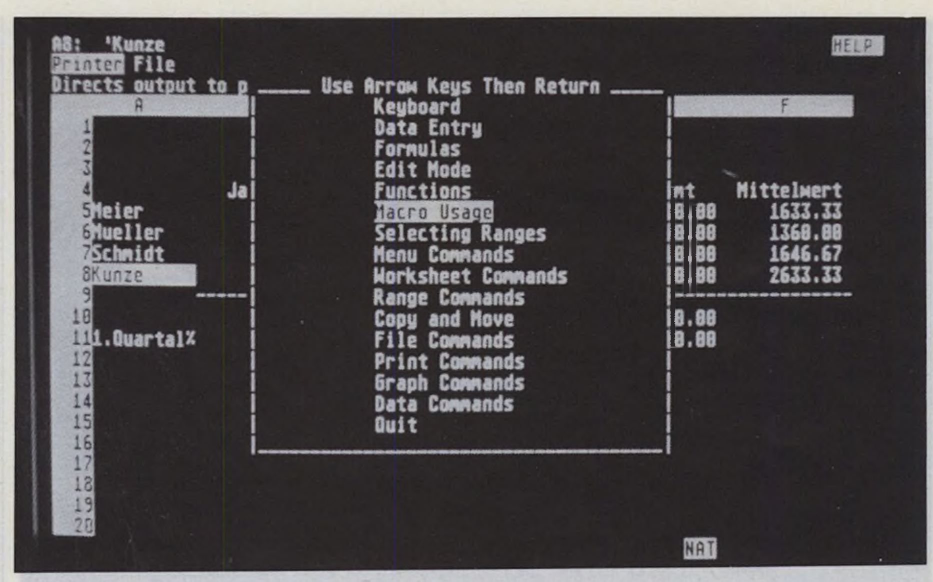
Das Arbeitsblatt – das Worksheet

Das Arbeitsblatt hat beim 520 ST+ eine Größe von 8 192 Zeilen ★ 230



Das Arbeitsblatt mit dem Hauptmenü

Spalten. Die Spaltenbreite ist von 1 – 72 Zeichen einstellbar. Die Zeilen werden durch die Zahlen 1 – 8 192 und die Spalten durch die Buchstaben A – IV bstimmt. Der Schnittpunkt von Zeile und Spalte bezeichnet eine einzelne Arbeitszelle. Man könnte also theoretisch eine maximale Anzahl von 1 884 160 Zellen verwalten. Dies wäre natürlich selbst mit einem Megabyte Speicherplatz nicht möglich. Hier wurde vielleicht schon an eine eventuelle Speichererweiterung gedacht. Eine Zeile kann mit numerischen oder alphanumerischen Daten bzw. Formeln belegt werden. Leider muß auch bemerkt werden, daß auch nicht durch Daten belegte Zellen vom Arbeitsspeicher mitverwaltet werden, so daß nicht die gesamte Worksheet-Kapazität zur Verfügung steht. Die einzelnen Zellen lassen sich außerdem kopieren, verschieben, schützen und vieles mehr.



Das Help-Menü

Anwendungsmöglichkeiten

Mit VIP Professional lassen sich so ziemlich alle, im täglichen Geschäftsleben anfallenden, Aufgaben erledigen. So sind Budgetplanungen, Marktanalysen, Bilanzvoraussagen und komplette Buchhaltung nur einige der vielfältigen Anwendungsmöglichkeiten.

Zum Beispiel ist eine „was ist wenn"-Analyse möglich, d. h. durch Ändern einer Umsatzzahl kann man schnell den Überblick über eine plötzlich eingetretene Veränderung der Geschäftssituation erhalten. Über die Funktionstaste F10 ruft man dann eine der eingebauten Standard-Business-Grafiken auf und erhält sogleich seine Daten in grafisch aufbereiteter Form auf dem Bildschirm angezeigt.

Als Standard-Grafiken verfügt VIP-Professional über Linie, Balken, Stapelbalken, Kreis und XY-Achse. Maximal sind sechs Datensätze in einer Grafik darstellbar. Bei der Grafik sind diverse Optionen, wie zum Beispiel das Versehen mit einem Titel, möglich. Einige der Beispiele kann man aus unseren Bildern entnehmen.

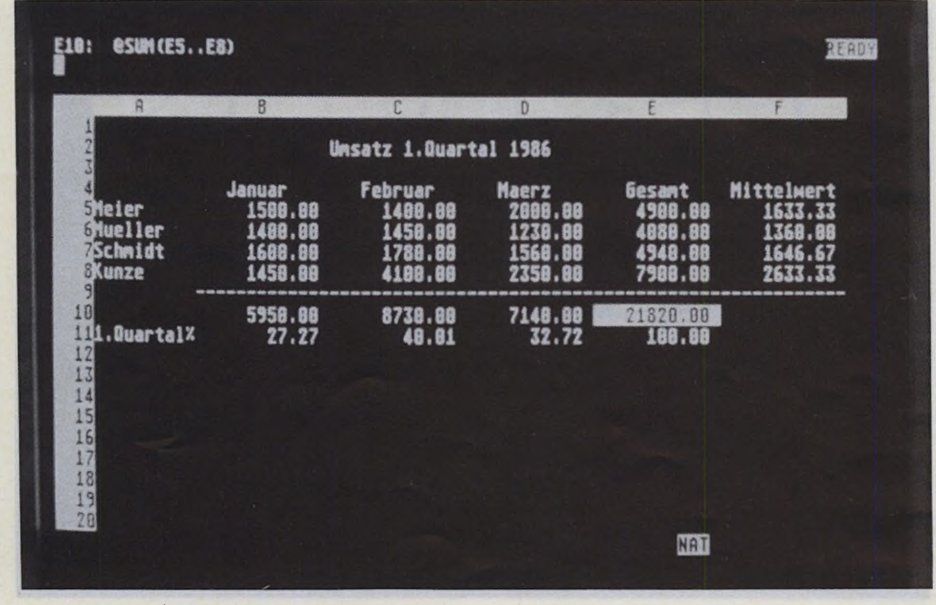
Zum VIP-Hauptprogramm gehört noch ein zusätzliches Druckprogramm für die mit VIP Professional erstellten Grafiken. Mit ihm lassen sich diverse Änderungen, wie zum Beispiel andere Schriftarten bei den Titeln, vornehmen. Dieses Programm arbeitet schon voll GEM-unterstützt.

Auch die Druckmöglichkeiten des Hauptprogrammes können sich sehen lassen. Es besteht zum Beispiel die Möglichkeit, jegliche Art von Beschriftungen mit dem Arbeitsblatt zusammen auszudrucken. Vorher kann man die Bereiche auswählen, die zu Drucker oder als Druckdatei zur weiteren Bearbeitung auf Diskette geschickt werden sollen.

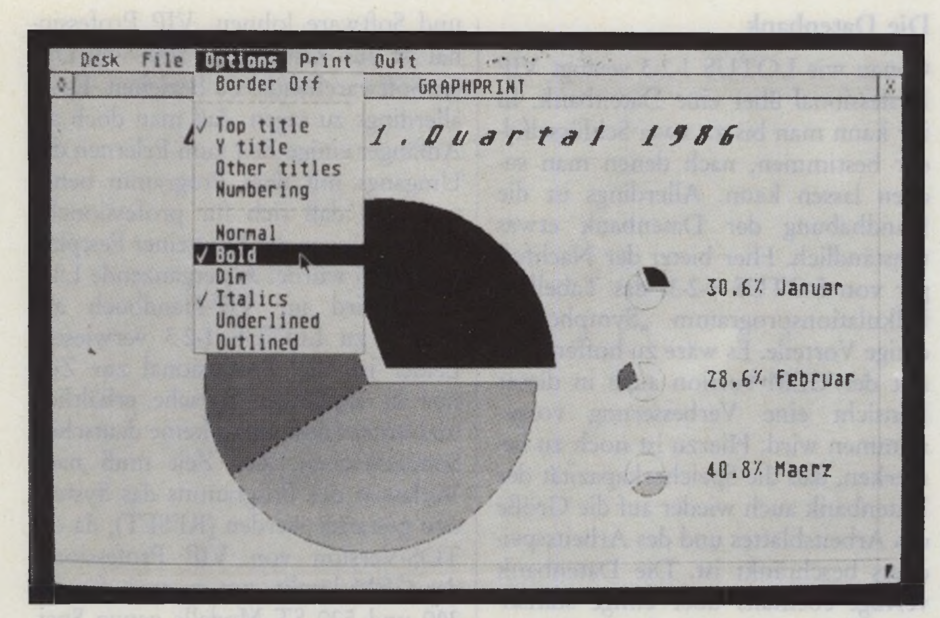
Verschiedene Datenformate

VIP Professional unterstützt auch den Zugriff auf Zweitlaufwerke und Disketten-Unterverzeichnisse. Man gibt dazu einfach den entsprechenden Laufwerks- oder Ordnernamen an.

Um das Arbeitsblatt abzuspeichern, stehen drei verschiedene Datenformate zur freien Wahl. Das erste ist das sogenannte WKS-Format, das von LOTUS 1-2-3 benutzt wird. Das zweite ist das NAT-Format, das man normalerweise auf dem ST dem WKS-Format vorziehen wird. Das dritte ist schließlich das ASCII-Format, mit dem man Arbeitsblätter abspeichern und dann später zum Beispiel in einem Textverarbeitungsprogramm weiterverarbeiten kann. Doch zunächst möchten wir noch kurz auf die anderen beiden Formate eingehen. Sie sind beide notwendig, die die ATARI ST Modelle ein anderes mathematisches Format als der IBM PC und seine Kompatiblen aufweisen. Wenn man nun ein mit VIP Professional erstelltes Arbeitsblatt auf



Ein gefülltes Arbeitsblatt



Das Druckprogramm arbeitet schon mit GEM

einem IBM PC benutzen will, muß man es im WKS-Format abspeichern. Dazu wird das ST-eigene NAT-Format in WKS umgewandelt. Diese Umwandlung braucht natürlich auch mehr Zeit, so daß man, wenn man nicht andauernd beide Rechnertypen einsetzen will, doch das schnellere NAT-Abspeicherungsformat wählen wird.

Vordefinierte Formeln

VIP Professional bietet unter anderem eine Vielzahl von schon vordefinierten Formeln und Verknüpfungen, von denen die wichtigsten im folgenden aufgeführt werden.

Finanz-mathematische Funktionen

IRR	– Interne Rücklaufrate
NPV	– momentaner Nettopreis
FV	– zukünftiger Preis
PV	– momentaner Preis
PMT	– Hypothekenzahlungen

Logische Funktionen

FALSE	– logische „0“
TRUE	– logische „1“
ISNA	– logische „1“, wenn kein Zugriff auf den Wert möglich ist
ISERR	– logische „1“, wenn der Wert nicht definiert
IF	– logische „1“ wenn eine Bedingung erfüllt ist.

Mathematische Funktionen

ACOS	– Arcus Cosinus
ASIN	– Arcus Sinus
ATAN	– Arcus Tangens im Bogenmaß
ATAN2	– Arcus Tangens
COS	– Cosinus
PI	– Zahl PI (3,14...)
SIN	– Sinus
TAN	– Tangens
ABS	– Absolutwert
EXP	– Exponent
INT	– ganzzahliger Wert
LN	– natürlicher Logarithmus
LOG	– Zehner-Logarithmus
MOD	– Rest einer Integerdivision
RAND	– Zufallszahl zwischen 0 und 1
ROUND	– Auf- bzw. Abrunden
SQRT	– Wurzel

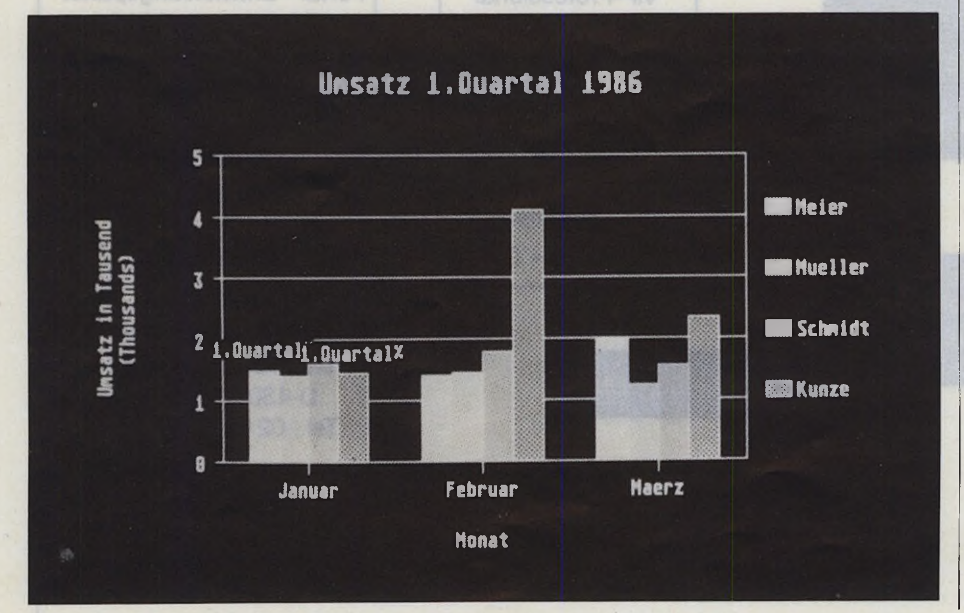
Statistische Funktionen

COUNT	– Zeigt alle Werte einer Liste
SUM	– Addiert alle Werte einer Liste
AVG	– Mittelwert
MIN	– Minimum
MAX	– Maximum
STD	– Standardabweichung
VAR	– Varianz

Das Programm bietet zusätzlich noch einige spezielle Funktionen, wie zum Beispiel Datum, Monat, Jahr und Fehleruntersuchungsroutinen.

Verarbeitung von Makros

VIP Professional bietet die Möglichkeit, sogenannte Makros zu programmieren. Dies sind vom Anwender erzeugte Prozeduren, die ganze Befehlsfolgen enthalten. Dazu konstruiert man das Makro auf einem leeren Arbeitsblatt und gibt ihm einen Namen. Später kann man es dann jederzeit aufrufen. Um den Ablauf von Makros während der Programmierung zu testen, ist sogar eine Abarbeitung im Einzelschritt-Modus möglich. Man kann also eine Makro-Prozedur Schritt für Schritt durch Tastendruck ablaufen lassen.



Balkengrafik: Eine der vielen Standardgrafiken von VIP Professional

Ferner verfügt VIP Professional über acht /X-Befehle. Diese Befehle können nur in Makros verwendet werden und dienen zur besseren Programmierung der Prozeduren.

Liste der /X-Befehle

- /XI – entspricht einer IF...THEN-Abfrage.
- /XG – springt zu einer bestimmten Zelle und fährt mit der Abarbeitung des Makros fort.
- /XC – springt zu einer bestimmten Zelle und ruft ein Unterprogramm auf.
- /XR – kehrt von einem Unterprogramm zurück.
- /XQ – beendet die Makroabarbeitung.
- /XM – gibt ein vorher definiertes Menü auf dem Bildschirm aus und wartet auf eine Eingabe.
- /XL – gibt einen Text auf dem Bildschirm aus und wartet auf eine Texteingabe.
- /XN – gibt einen Text auf dem Bildschirm aus und wartet auf eine Zahleneingabe.

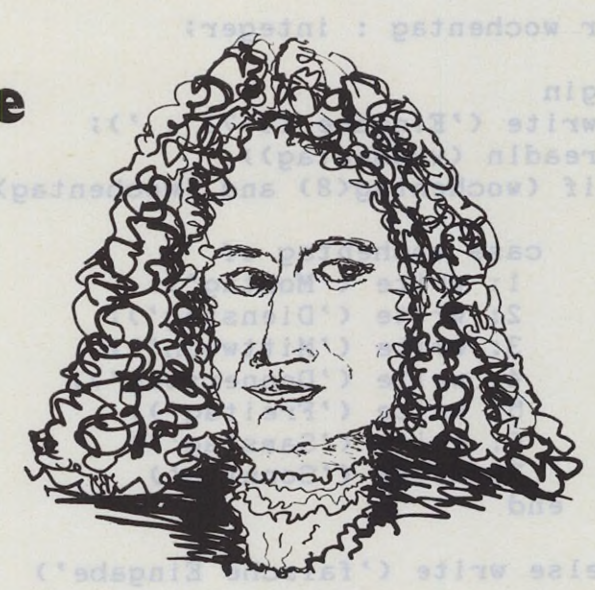
Die Datenbank

Genau wie LOTUS 1-2-3 verfügt VIP Professional über eine Datenbank. In ihr kann man bis zu zwei Schlüsselfelder bestimmen, nach denen man suchen lassen kann. Allerdings ist die Handhabung der Datenbank etwas umständlich. Hier bietet der Nachfolger von LOTUS 1-2-3, das Tabellenkalkulationsprogramm „Symphony“, einige Vorteile. Es wäre zu hoffen, daß mit der GEM-Version auch in dieser Hinsicht eine Verbesserung vorgenommen wird. Hierzu ist noch zu bemerken, daß die Speicherkapazität der Datenbank auch wieder auf die Größe des Arbeitsblattes und des Arbeitsspeichers beschränkt ist. Die Datenbank verfügt ebenfalls über einige statistische Funktionen, die denen des Arbeitsblattes entsprechen.

Was uns am VIP Professional gefallen hat, ist die hervorragende Tabellenkalkulation mit vielen vordefinierten Formeln und die sehr gute Grafik. Weiterhin positiv zu bewerten ist die Kompatibilität zu IBM - LOTUS 1-2-3, so daß Anwender mühelos auf den ATARI umsteigen können. Dies würde sich allein schon vom deutlich niedrigeren Anschaffungspreis für Hard- und Software lohnen. VIP Professional ist zur Zeit für ca. 500–750 DM im Softwarehandel zu beziehen. Es ist allerdings zu sagen, daß man doch als Anfänger einige Zeit zum Erlernen des Umgangs mit dem Programm benötigt, und daß sich für professionelle Anwendungen der Kauf einer Festplatte lohnen würde. Als ergänzende Literatur wird im VIP-Handbuch auf Bücher zu LOTUS 1-2-3 verwiesen. Leider ist VIP Professional zur Zeit nur in englischer Sprache erhältlich und unterstützt somit keine deutschen Sonderzeichen. Zur Zeit muß nach Verlassen des Programms das System neu gestartet werden (RESET), da die TOS-Version von VIP Professional das GEM löscht, um so auch für die 260 und 520 ST Modelle genug Speicherplatz zur Verfügung stellen zu können. Dies wird aber mit der GEM-Version, die gegen Einsendung der Originaldiskette und einen Unkostenbeitrag von DM 39,– DM im 24-Stunden-Service umgetauscht wird, sicher entfallen.

LOTUS 1-2-3 und Symphony sind eingetragene Warenzeichen der LOTUS Development GmbH.

Einführung in die Programmiersprache PASCAL Teil 2



Im ersten Teil dieses Kurses wurden schon einige wichtige Eigenschaften von Pascal besprochen. So zum Beispiel der Programmaufbau und der Vereinbarungs- und Anweisungsteil. Außerdem wurden schon folgende grundlegende Elemente behandelt:

Label
Konstanten
Variablen
– Zuweisung
read, readln
write, writeln
– formatierte Ausgabe
sqr(x), sqrt(x)
begin ... end
if ... then ...
if ... then ... else ...

Zum Erstellen von Programmen wurde das Struktogramm (auch Nassi-Shneiderman-Diagramm) eingeführt. Es erleichtert das 'saubere' bzw. strukturierte Programmieren und soll im weiteren Kurs ausgebaut werden. Ein weiteres Hilfsmittel für die Programmierung in Pascal sind die **Syntax-Diagramme**. Sie zeigen eindeutig die Syntax aller Elemente und Strukturen der Sprache an. Für die ersten Programme sind sie nicht notwendig. Sie werden aber in einem der nächsten Abschnitte des Kurses eingeführt und erläutert.

Wie schon im ersten Teil gesagt, werden die Programme mit TURBO-Pascal erstellt.

Die Programme laufen aber (falls nicht anders angegeben) auch auf anderen Pascal-Compilern.

Doch nun geht es richtig los: Wenn man an einer Stelle viele verschiedene Fälle unterscheiden muß, so kann man dies mit einer geschachtelten if-then-else-Anweisung machen. Es gibt dafür jedoch einen viel leistungsfähigeren Befehl, die **CASE-Anweisung**. Sie erlaubt es, mehrere Fälle gezielt anzuspringen. Die Anweisung hat die Form:

```
CASE Ausdruck OF
     Konstante 1 : Anweisung 1;
     Konstante 2 : Anweisung 2;
         ...
     Konstante n : Anweisung n
END
```

Die verschiedenen Anweisungen können dabei übersichtlich untereinander geschrieben werden. Dadurch ergibt sich ein anschaulicheres Bild als bei einer geschachtelten if-then-else-Anweisung. Dies kann man auch am Struktogramm erkennen, das in Bild 1 dargestellt wird.

Das Programm (Listing 1) erwartet als Eingabe eine Ganzzahl zwischen eins und sieben. Es wird dann der entsprechende Wochentag ausgegeben. Wenn Sie das Listing eingegeben haben, sollten Sie sich damit ein wenig beschäftigen. Probieren Sie einige Werte aus, auch solche, die nicht im angegebenen Bereich liegen.

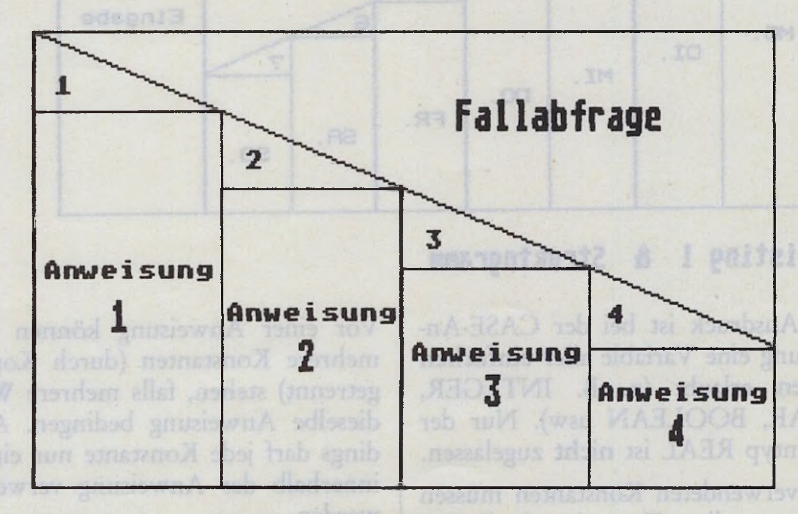


Bild 1 : CASE - Anweisung

```
program WOCHENTAG;

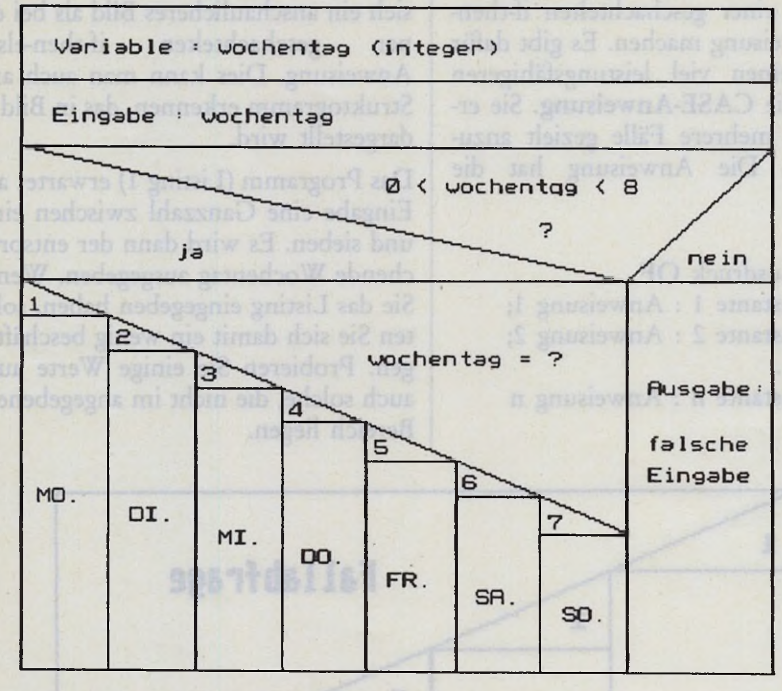
var wochentag : integer;

begin
  write ('Eingabe (1-7) : ');
  readln (wochentag);
  if (wochentag<8) and (wochentag>0) then

     case wochentag of
      1: write ('Montag');
      2: write ('Dienstag');
      3: write ('Mittwoch');
      4: write ('Donnerstag');
      5: write ('Freitag');
      6: write ('Samstag');
      7: write ('Sonntag')
     end

  else write ('falsche Eingabe')

end.
```



Listing 1 & Struktogramm

Als Ausdruck ist bei der CASE-Anweisung eine Variable aller **einfachen Typen** erlaubt (z. B. INTEGER, CHAR, BOOLEAN usw). Nur der Datentyp REAL ist **nicht** zugelassen.

Die verwendeten Konstanten müssen alle vom selben Typ sein wie die Variable.

Vor einer Anweisung können auch mehrere Konstanten (durch Komma getrennt) stehen, falls mehrere Werte dieselbe Anweisung bedingen. Allerdings darf jede Konstante nur einmal innerhalb der Anweisung verwendet werden.

Wichtig ist bei der CASE-Anweisung außerdem, daß für alle Werte, die die Variable annehmen kann eine entsprechende Konstante vorgegeben wird. Sollte dies nicht der Fall sein, so tritt ein Fehler auf. Um diese Fehlerquelle zu umgehen, kann man die Variable vor ihrer Benutzung in der CASE-Anweisung auf ihren Bereich hin kontrollieren. Dies wird mit einer einfachen **if**-Abfrage realisiert (siehe Listing 1 und Struktogramm). Mit einem zugefügten **else** Zweig kann man dann alle anderen Fälle behandeln.

Bei **TURBO-Pascal** ist das ELSE bereites als fester Bestandteil der CASE-Anweisung (siehe Bild 2) vorgesehen, dadurch entfällt die Abfrage mit IF. In manchen Pascal-Dialekten gibt es den Befehl **OTHERS**. Er ersetzt dann das else. Eine vorangehende Abfrage mit 'if' ist auch hier nicht notwendig. Bei **UCSD-Pascal** und auch bei **TURBO-Pascal** tritt keine Fehlermeldung auf, wenn der Wert der Variablen keiner Konstanten entspricht. Es wird in diesem Fall mit der Programmanweisung, die hinter dem CASE-Block steht, fortgefahren.

Auffallend bei der CASE-Anweisung ist, daß sie mit einem END abgeschlossen werden muß, ohne ein BEGIN am Anfang zu benötigen.

Ein weiteres Beispiel für die CASE-Anweisung ist in Struktogramm 2 dargestellt. Bei diesem Programm (Listing 2) wird nach Eingabe eines Buchstabens ein Name mit diesem Buchstaben ausgegeben. Hier wird erstmals der Datentyp **CHAR** verwendet. Die Variable 'buchstabe' kann nun alphanumerische Zeichen annehmen. In diesem Fall sind jedoch nur die Buchstaben des Alphabets relevant. Sie werden im Programm immer in Hochkommas (z. B. buchstabe: = 'a') gestellt, um sie von Variablen unterscheiden zu können.

Zur Veranschaulichung sind hier nur die Marken 'a', 'b', 'c', 'd' und 'k' belegt. Bei allen anderen wird der Text 'kein Name gespeichert!' ausgegeben. Zu diesem Zweck wird die Eingabe mit der **IN**-Operation kontrolliert. Sie hat die Form:

Variable **IN** [Menge]

```
program NAMEN;

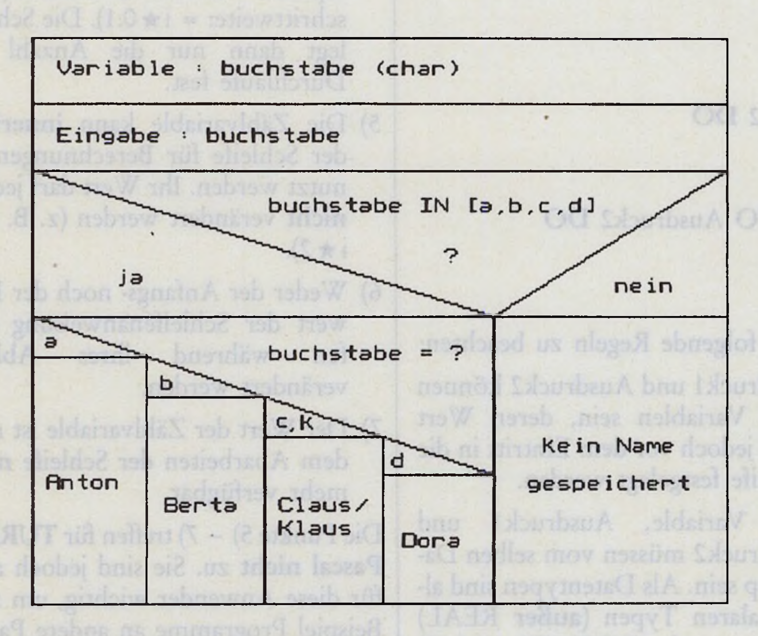
var buchstabe : char;

begin
   write ('Eingabe (a-z) : ');
   readln (buchstabe);
   if buchstabe in ['a','b','c','d','k'] then

   case buchstabe of
   'a'    : write ('Anton');
   'b'    : write ('Berta');
   'c','k': write ('Claus/Klaus');
   'd'    : write ('Dora')
   end

   else   write ('kein Name gespeichert!');

end.
```



Listing 2 & Struktogramm

Dabei muß natürlich der Datentyp der Variable mit dem der Menge übereinstimmen. Da hier eine Menge von Buchstaben benötigt wird, muß die Variable als CHAR definiert sein.

Achtung: die IN-Funktion macht, wie auch die CASE-Anweisung, eine genaue Unterscheidung zwischen Groß- und Kleinbuchstaben. Man kann verzweifeln, wenn die Caps Lock Taste gedrückt ist und das Programm nicht erwartungsgemäß reagiert.

Das Ergebnis ist vom Type BOOLEAN. Dieser Typ ist schon von anderen IF-Anweisungen her bekannt (siehe z. B. Listing 1) oder auch von der CASE-Anweisung im nächsten Programm. Das bedeutet, daß die Aussage die 'Werte' false (falsch) und true (wahr) annehmen kann, je nachdem ob die Aussage wahr (Variable ist Teilmenge von Menge) oder falsch ist (Variable ist **nicht** Teilmenge von Menge).

```
program ENTSCHEIDUNG;

var entscheidung : boolean;
    zahl         : real;

begin
  write   ('--- zahl>0 ---     ');
  write   ('Eingabe (1,0,-1): ');
  readln (zahl);

  case zahl>0 of
   true : write ('--- wahr');
   false: write ('--- falsch');
  end;

end.
```

Das dritte Beispiel zur CASE-Anweisung (Listing 3) setzt als Ausdruck den gerade besprochenen Typ Boolean ein. Als Konstanten bzw. Marken werden die Werte 'true' und 'false' eingesetzt. Je nachdem ob nun der Ausdruck 'zahl>0' wahr oder falsch ist, wird zum entsprechenden Teil verzweigt.

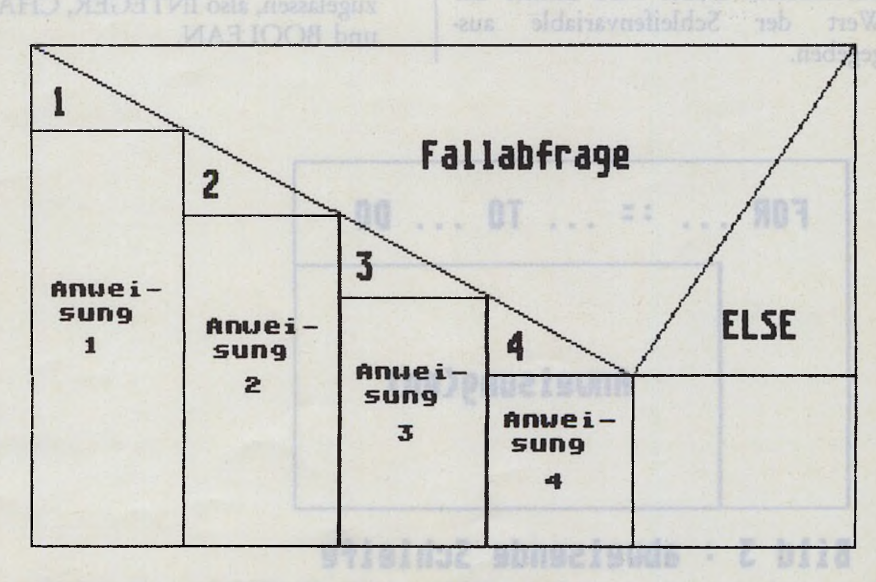


Bild 2 : CASE/ELSE - Abfrage

Im nun folgenden Teil wollen wir uns mit den Möglichkeiten beschäftigen, die Pascal zur **Schleifensteuerung** anbietet. Diese Steuerbefehle sind für die Programmierung von großer Bedeutung, weil sie es ermöglicht, eine Anweisung oder eine Folge von Anweisungen so lange zu **wiederholen**, bis eine vorher definierte Abbruchbedingung eintritt. Dies ist zwar prinzipiell auch mit einer IF-Abfrage und einem anschließenden GOTO-Befehl erreichbar, doch ist diese Art der Schleifenbildung sehr unübersichtlich und fehleranfällig. Bei Pascal hat man die Auswahl zwischen drei verschiedenen Schleifensteuerbefehlen. Dabei hat jeder Befehl seine Besonderheiten und somit seinen speziellen Anwendungsbereich. Die bekannteste Schleifenanweisung hat die Form:

```
program FOR_Schleife;
var i : integer;
begin
    for i:=1 to 10 do
        writeln (i)
end.
```

FOR Variable : = Ausdruck1 TO Ausdruck2 DO

bzw.

FOR Variable : = Ausdruck 1 DOWNTO Ausdruck2 DO

Diese Anweisung ist der aus Basic bekannten FOR-NEXT-Schleife ähnlich. Ihr Struktogramm wird in Bild 2 dargestellt. Bei 'DOWNTO' zählt die Schleife rückwärts.

Die Anzahl der Schleifenwiederholungen liegt in diesem Fall von Anfang an fest. So wird zum Beispiel die Schleife im Programm (Listing 4) zehnmal durchlaufen. Dabei wird immer der Wert der Schleifenvariable ausgegeben.

Es sind folgende Regeln zu beachten:

1) Ausdruck1 und Ausdruck2 können auch Variablen sein, deren Wert **muß** jedoch vor dem Eintritt in die Schleife festgelegt werden.
2) Die Variable, Ausdruck1 und Ausdruck2 müssen vom selben Datentyp sein. Als Datentypen sind alle skalaren Typen (**außer** REAL) zugelassen, also INTEGER, CHAR und BOOLEAN.
3) Wenn der Endwert schon am Anfang größer ist als der Anfangswert (bzw. umgekehrt bei DOWNTO), so wird der Schleifeninhalt **nicht** bearbeitet. Das Programm läuft dann mit der Anweisung, die hinter dem Schleifenblock steht, weiter. Man nennt diese Form deshalb auch **abweisende Schleife** (siehe auch Bild 3).
4) Die einzig mögliche Schrittweite ist 1 bzw. – 1. Diese Festlegung hat neben ihrer Einschränkung den Vorteil, daß bei einer Erhöhung der Schrittweite um Eins kein Rundungsfehler auftreten kann, wie dies in Basic häufig der Fall ist. Wenn man eine andere Schrittweite benötigt, so **muß** man dafür eine separate Variable verwenden (z. B. schrittweite: = i★0.1). Die Schleife legt dann nur die Anzahl der Durchläufe fest.
5) Die Zählvariable kann **innerhalb** der Schleife für Berechnungen benutzt werden. Ihr Wert darf jedoch **nicht** verändert werden (z. B. i: = i★2).
6) Weder der Anfangs- noch der Endwert der Schleifenanweisung dürfen während ihres Ablaufs verändert werden.
7) Der Wert der Zählvariable ist nach dem Abarbeiten der Schleife **nicht** mehr verfügbar.

Die Punkte 5) – 7) treffen für **TURBO-Pascal nicht** zu. Sie sind jedoch auch für diese Anwender wichtig, um zum Beispiel Programme an andere Pascal-Versionen anpassen zu können. Wenn man die Einschränkungen, die für fast alle Pascal-Versionen gelten, nicht kennt kann es zu bösen Überraschungen kommen, wenn man auf einmal mit ihnen programmieren muß.

Das Beispielprogramm (Listing 5) verdeutlicht die Eigenarten von TURBO-Pascal. Es ist, in dieser Form, sonst **nicht** lauffähig!

Ansonsten stellt das Programm eine häufig verwendete Form dar, nämlich eine **geschachtelte Schleife**. Schleifen können in beliebiger Anzahl ineinander verschachtelt werden. Sie dürfen sich dabei jedoch nicht überschneiden. Dies wäre bei der Darstellung mit Hilfe des Struktogramms auch nicht möglich (siehe Struktogramm 5).

FOR ... := ... TO ... DO

Anweisung(en)

Bild 3 : abweisende Schleife

```
program FOR_TurboPascal;

var i,ende : integer;

begin
     ende:=10;
     for i:=1 to ende do
        begin
        writeln;
        write (i:2);
        i:=i+1;
        write (i:4);
        for i:=1 to i do
           write ('*');
        end;
     writeln;
     write ('i nach Schleifenende ---> ',i)
end.
```

Listing 5

Programm FOR_TurboPascal
Variablen : i , ende (integer)
ende := 10
FOR i := 1 TO ende DO
Ausgabe : i
i := i + 1
Ausgabe : i
FOR i := 1 TO i DO
Ausgabe : *
Ausgabe : i nach Schleifenende

Struktogramm 5

Wenn Sie sich das Listing näher betrachten, werden Sie feststellen, daß hier gegen viele Regeln aus 'Standard-Pascal' verstoßen wird. So wird z. B. der Wert der Zählvariable bei jedem Durchgang innerhalb der Schleife zusätzlich um Eins erhöht. Trotzdem kommt die äußere Schleife nicht aus dem 'Tritt'. Sie läuft, wie am Anfang festgelegt 10 mal durch, obwohl i schon viel früher die Endbedingung (i = 10) erreicht hat (siehe Probelauf Bild 4).

```
Compilation aborted
>

Compiling
  20 lines

Code:    212 bytes (8257-832B)
Free:  25294 bytes (832C-E5FA)
Data:     11 bytes (E5FB-E606)

Running

 1   2**
 3   4****
 5   6******
 7   8********
 9  10**********
11  12************
13  14**************
15  16****************
17  18******************
19  20********************
i nach Schleifenende ---> 20
>
```

Bild 4: Probelauf

Auch die innere Schleife ist beachtenswert, denn es wird zum Zählen und für die Endbedingung dieselbe Variable i verwendet. Bei TURBO-Pascal spielt das keine Rolle. Die Endbedingung wird vor der Ausführung der Schleife festgelegt und kann dann durch nichts mehr geändert werden.

Auch nach dem Durchlaufen der Schleife ist die Zählvariable noch verfügbar, das heißt, ihr Wert kann ausgedruckt oder weiterverarbeitet werden.

Im nächsten Teil geht es weiter mit den Kontrollstrukturen. Es wird die Struktur der REPEAT/UNTEIL- und der WHILE/DO-Anweisung besprochen.

HARDWARE

Achtung Superpreise
Disketten 1D 135 TPI 7,40
3 1/2" 2D 135 TPI 9,00
Fa. Gaudek + Bublak 8045 Ismaning, Konradstr. 7, 089/96 71 37

Günstige Laufwerke:
BASF 6164, 3.5", 1 MB, 349,–
BASF 6138, 5.2", 1 MB, 359,–
Tel. 0 89/80 68 23, 18–22 Uhr

Den ATARI aufrüsten mit: DRAMs NEC D41256C-150 ns
5– 9 Stück DM 8,88
10–19 Stück DM 8,55
20–49 Stück DM 8,22
50–99 Stück DM 7,89
Computertechnik Klepsch 5828 Ennepetal; Telefon:
0 23 33-8 02 02 9–12/14–17 h

Bin 100 % schwerbeh. m. kl. Rente. Wo kann ich kompl. 260 o. 520 ST + Anlage preisw. eink. Tel. 0 62 01/6 68 88

520 ST Atari Laufw. SF 354 neuw. **450,–.** 0 23 92/78 78

ATARI SF-354 Diskettenlaufwerk mit Kabel und Netzteil für nur 395,– DM zu verkaufen. Tel.: 0 61 92/88 85

Softy Hard's Computershop Wir sind die **ATARI ST Spezialisten** rue de Lausanne 28/1700 Freiburg CH

SF 354/Garantie für 400,– Siems 02 21/79 84 22 ab 14 h

Superangebot: 10 Disketten voll mit Software im Wert von 2 000,– DM, 520 ST, SF 354, SM 124, Maus, alles zusammen nur 3 000,– DM. 02 01/27 28 51

SOFTWARE

VIP Professional
(Lotus 1,2,3), 480,– DM,
Tel. 02 21/4 30 14 42

Charakter-Analyse für ST
7 Seiten Auswertung.
Programmiert unter **GEM!**
DISK DM 69,–, Info gratis!
TJ-Soft, Pf. 65, 3250 Hameln

ATARI ST – Biorhythmus mit Supergrafik. Voll Maus-unterstützt bietet dieses Programm phantastische Bedienerfreundlichkeit. Grafikausdruck! Auf Disk: 49,– DM
Pahlen & Kraus, Software, Kolonnenstr. 28, 1000 Berlin 62, Tel. 0 30-7 82 69 06

Verschiedene Hilfsprogr. und **Pbl.D.-Soft:** Disk + 1,10 o. –,50 für Liste an: Rose, A.-Flend.-Str. 284, 4290 Bocholt

Flip Side **79,– Seka** Assembler 169,– **Skull/West . 119 Mitchtron 79,– Havoc 79,–** N. Brandes Salzdahlumer 60 33 BS Liste –,80 NN + 4,–

UMSONST
gibt es bei uns den neuen ST-Softwarekatalog. Postkarte an: STWARE, Ettenhofenerstr. 31, 8031 Weßling

Atari 260/520 ST
Riesenangebot an Software
Sofort Katalog anfordern:
STWARE, Ettenhofenerstr. 31,
8031 Weßling

Wer sagt denn es gäbe für ATARI ST keine Software? Größter Katalog geg. 2,– DM in Briefm. von **Brain Wave.** 56 WPT-1. Postfach 13 06 24

ST Software für s/w Monitor gesucht! Liste an: M. Klocke-Sewing, Gronauer Str. 19, 4800 Bielefeld 1

Bin 100 % schwerbeh. m. kl. Rente. Suche preisw. Prog. für ST-Anlage. Wer kann helfen. Tel. 0 62 01/6 68 88

ATARI ST Clubsoftware, Public Domain Software, für den kaufmännischen u. privaten Bereich. Tel. 0 61 44/17 38

Ich, Besitzer eines 520ST+, bin in der Computertechnik nicht bewandert. Wer hilft mir? Evtl. auch Programm erstellen. Tel. 0 89/90 26 61

KONTAKTE
Suche ST-Kontakte in Ostfriesland. Günter Tholen, 2970 Emden, Tel: 0 49 21/2 06 12

Wer gibt Computer-Neuling Einweisung in ST? Raum 7750 KN T. 0 75 31/5 56 77

Programmierer gesucht für A. ST. Wer hat 68000 Basic, das kompatibel zu C 64, MBasic oder GWBasic ist. Braun, Wildermuthring 21, 2 Hamburg 62

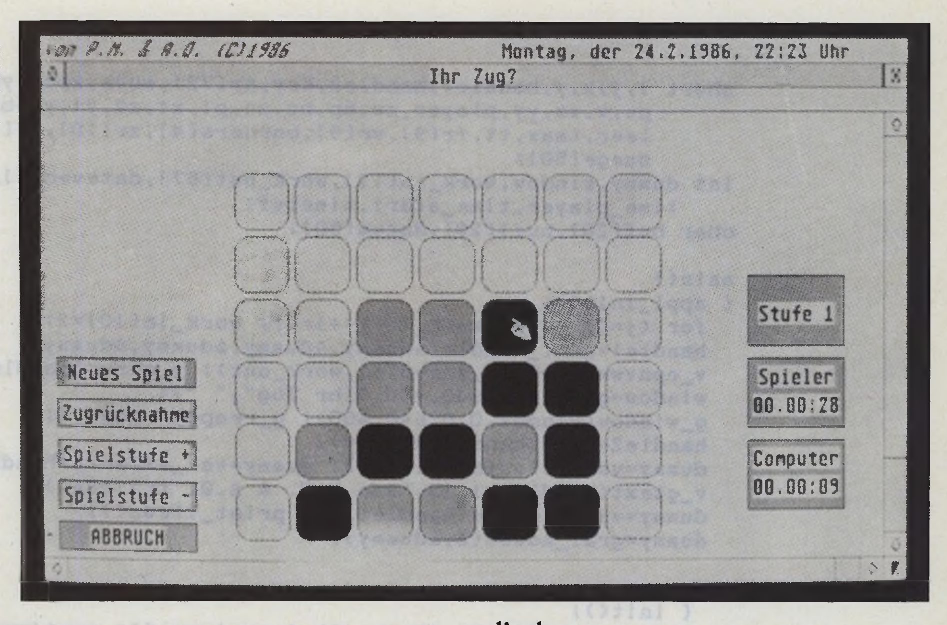
Raum Lev. Suche User zweck Erfahrungsaustausch.
5090 Lev1 an D. Dingbank 27

TAUSCH

PRG in BASIC, C, Modula-2 tauscht A. Will, Leo-Weismantelstr. 1B, 87 Würzburg.

4 in einer Reihe

Ein Denkspiel in "C"



Dieses Spiel wird in einer senkrechten Ebene von 7 horizontalen und 6 vertikalen Feldern gespielt. Die Spieler setzen abwechselnd einen Stein auf eine bestimmte Spalte. Der Stein fällt daraufhin auf das tiefste unbesetzte Feld herunter. Derjenige Spieler, dem es gelingt, zuerst vier Steine horizontal, vertikal oder diagonal miteinandern zu verbinden, hat gewonnen.

In dieser Programmversion tritt der Rechner als Spielpartner an. Der Dialog zwischen Spieler und ATARI findet über die Titelzeile des Spielfensters, dessen Infozeile und die Maus statt. In der oberen rechten Bildschirmecke wird die Echtzeituhr samt Datumsangabe angezeigt. Es empfiehlt sich daher, dieselbe vor dem Laden des Programms einzustellen. Während der Bedenkzeit des Computers wird diese Ausgabe nicht aktualisiert, um die Zugzeiten kurz zu halten. Rechts neben dem Spielfeld werden die Bedenkzeiten des Spielers und des Rechners und die aktuelle Spielstufe angezeigt. Das Programm ist vollständig in „C" geschrieben.

Bedient wird das Programm ausschließlich mit der Maus. Folgende Funktionen stehen durch einfaches Anklicken zur Verfügung:

- Auswahl der zu besetzenden Spalte
- Beginn eines neuen Spielers
- stufenlose paarweise Zugrücknahme
- erhöhen der Spielstärke
- verringern der Spielstärke
- Abbruch des Programms mit Sicherheitsabfrage; nur wenn die Taste 'j' gedrückt wird, bricht das Programm ab

Es stehen fünf verschiedene Spielstärken (1 bis 5) zur Verfügung, wovon 1 die leichteste ist. Die Zugzeiten zu Spielbeginn differieren erheblich. Sie liegen zwischen einer und 400 Sekunden. Mit abnehmender Anzahl der noch zu besetzenden Spalten allerdings, nimmt die Zugzeit des Rechners erheblich ab.

Zu Beginn des Programmes werden sämtliche Variablen global definiert. Es werden die Typen 'short' (16 Bit), 'int' (32 Bit) und 'char' (8 Bit) verwendet.

Nachstehend folgt eine Erläuterung der Aufgaben einzelner Funktionen:

main
Initialisierung von VDI, AES und einem Window; Hauptschleife für sämtliche Programmfunktionen.

nowinner
Unentschiedenes Spielende.

playerwins
Spielende bei Sieg des Spielers.

atariwins
Spielende bei Sieg des Rechners.

init
Initialisieren des Spielfeldes, der Zugzeitvariablen, etc; Ausgabe des Spielfeldes

entry
Überwachen der Maus; Aufruf der entsprechenden Routinen bei Auswahl einer Funktion.

depth
Ermitteln der ersten freien Zeilenposition innerhalb einer angewählten Spalte.

display
Ausgabe eines durchgeführten Zuges.

points
Ermitteln der Nachbarspielsteine für einen vom Computer geprüften Zug.

sub1500
Erhöhen der Bewertungspunktzahl in Abhängigkeit von der Anzahl der eigenen Nachbarspielsteine.

computer
Hauptroutine für den Rechnerzug; entweder wird der allererste Zug zufällig ausgewählt, oder aber die rekursive Funktion **rekursion** aufgerufen.

rekursion
Ermitteln des besten Zuges auf der Basis einer Zugtiefenschachtelung mit Punktebewertung

print_items
Ausgabe der anwählbaren Funktionen im Window.

get_time
Holen der aktuellen Uhrzeit und Umwandlung in sec.

show_time
Anzeigen der spielerspezifischen Bedenkzeit.

print_max
Ausgabe der Spielstufe.

test
Ermittlung der Anzahl noch besetzbarer Spalten.

show_date
Ausgabe der Echtzeituhr mit Tagesdatum.

abbruch
Sicherheitsabfrage bei Auswahl der Abbruch-Funktion.

```
short i,j,x,y,handle1,handle2,key,ar[72],move,xpos,ypos,buttons,index,
      pu,w,xx,yy,player,zw,bp,bz,zu,p1,x1,x2,y1,y2,b2,p2,style,myexit,
      leer,tmax,tt,fr[9],wr[9],corners[4],xr[10],yr[10],br[10],pr[10],
      zuege[50];
int dummy,window,work_in[12],work_out[57],datevec[7],time,start,
    time_player,time_atari,timebuf;
char buf[20],buf1[20],datum[50];

main()
{ appl_init();
  for (j=0; j<10; work_in[j++]=1); work_in[10]=2;
  handle1=graf_handle(&dummy,&dummy,&dummy,&dummy);
  v_opnvwk(work_in,&handle1,work_out); v_clrwk(handle1);
  window=g_open(1,640,400,"Ihr Zug"," ");
  g_window(window,0,0,640,400); g_reopen(window);
  handle2=g_vdihandle(window);
  dummy=vst_color(handle1,1); dummy=vst_effects(handle1,4);
  v_gtext(handle1,1,15,"von P.M. & A.Q. (C)1986");
  dummy=vst_effects(handle1,0); print_items();
  dummy=graf_mouse(3,&dummy);

  do
   { init();
     do { test(); if (leer==0) { nowinner(); continue; }
          entry();
          if (myexit==0)
          { if (pu>=100) { playerwins(); continue; }
            test(); if (leer==0) { nowinner(); continue; }
            player=1; computer(); player=1; index=bz; depth();
            display(); points();
            if (pu>=100) { atariwins(); }
          }
        }
       while(myexit==0);
   }
   while(myexit==1);

  dummy=graf_mouse(0,&dummy);
  g_exit();
  v_clsvwk(handle1);
  appl_exit();
 }

nowinner()
{ g_info(window,"Sie haben Glck. Das Spiel endet unentschieden.");
  myexit=1; g_title(window,"Spielende, bitte Mausknopf drcken!");
  do { show_date(); vq_mouse(handle2,&buttons,&xpos,&ypos); } while(buttons==0);
}

playerwins()
{ g_info(window,"Welch bses Schicksal. Sie haben zufllig gewonnen!");
  myexit=1; g_title(window,"Spielende, bitte Mausknopf drcken!");
  do { show_date(); vq_mouse(handle2,&buttons,&xpos,&ypos); } while(buttons==0);
}

atariwins()
{ g_info(window,"Es tut mir fast leid: SIE HABEN VERLOREN!");


  myexit=1; g_title(window,"Spielende, bitte Mausknopf drcken");
  do { show_date(); vq_mouse(handle2,&buttons,&xpos,&ypos); } while(buttons==0);
}

init()
{ myexit=0; time_player=0; time_atari=0; tmax=2; zu=0;
  g_info(window," ");
  for (i=0; i<72; ar[i++]=99);
  for (i=10; i<=55; i+=9) { for (j=0; j<7; ar[i+j++]=0); }
  dummy=vsf_color(handle2,1); dummy=vsf_interior(handle2,2);
  vsf_perimeter(handle2,1); dummy=vsf_style(handle2,1);
  for (y=1; y<7; y++)
   { for (x=1; x<8; x++)
       { corners[0]=100+x*45; corners[1]=y*45;
         corners[2]=corners[0]+40; corners[3]=corners[1]+40;
         g_rfbox(window,corners);
       }
```

```
    }
   dummy=vsf_interior(handle2,2); print_max();
   player=1; show_time(); player= -1; show_time();
  }
  entry()
  { player= -1; move= -1; v_show_c(handle2,0);
    get_time(); start=time;
    do
     { g_title(window,"Ihr Zug?");
       do { show_date(); vq_mouse(handle2,&buttons,&xpos,&ypos); } while(buttons==
0);
       if (xpos>11 && xpos<113)
        { if (ypos>353 && ypos<375) { abbruch(); continue; }
          if (ypos>233 && ypos<255) { myexit=1; continue; }
          if (ypos>293 && ypos<315 && tmax<6) {tmax++; print_max(); continue; }
          if (ypos>323 && ypos<345 && tmax>2) {tmax--; print_max(); continue; }
          if (ypos>263 && ypos<285 && zu>0)
           { dummy=vsf_style(handle2,1);
             for (i=0; i<2; i++)
              { zu--; index=zuege[zu]; depth();
                if (move<0) move=1; else move++; ar[move*9+index]=0;
                corners[0]=100+index*45; corners[1]=move*45;
                corners[2]=corners[0]+40; corners[3]=corners[1]+40;
                g_rfbox(window,corners);
              }
             do { vq_mouse(handle2,&buttons,&xpos,&ypos); } while(buttons!=0);
             move= -1;
           }
        }
       if (xpos>144 && xpos<460 && ypos>60) { index=(xpos-100)/45; depth(); }
     }
    while(move<0 && myexit==0);
    v_hide_c(handle2); zuege[zu]=index; zu++;
    get_time(); time_player+=time-start; show_time();
    if (myexit==0) { display(); points(); }
  }
  depth()
  { move= -1; for (y=7; y>0; y--) { if (ar[y*9+index]==0) { move=y; break; }}
  }
  display()
  { ar[move*9+index]=player;
    if (player<0)
     style=4; else style=8; dummy=vsf_style(handle2,style);
     y=1; corners[0]=100+index*45; corners[1]=y*45;
     corners[2]=corners[0]+40; corners[3]=corners[1]+40;
     g_rfbox(window,corners);
    while(y<move)
     { delay(3); y++; dummy=vsf_style(handle2,1);
       corners[1]=(y-1)*45; corners[3]=corners[1]+40;
       g_rfbox(window,corners);
       dummy=vsf_style(handle2,style);
       corners[1]=y*45; corners[3]=corners[1]+40;
       g_rfbox(window,corners);
     }
    for (i=0; i<3; i++)
     { delay(10); dummy=vsf_style(handle2,1); g_rfbox(window,corners);
       delay(10); dummy=vsf_style(handle2,style); g_rfbox(window,corners);
     }
  }
  points()
  { pu=0;
    for (xx= -1; xx<2; xx++)
     { for (yy= -1; yy<2; yy++)
        { if (xx==0) { if(yy==0) continue; }
          w=1; fr[(xx+1)*3+yy+1]=0;
          while(ar[move*9+index+xx*w+(yy*9)*w]==player) { w++; }
          if (ar[move*9+index+xx*w+(yy*9)*w]==0) fr[(xx+1)*3+yy+1]=1;
          wr[(xx+1)*3+yy+1]=w-1;
        }
     }
    for (yy= -1; yy<2; yy++)
     { zw=wr[yy+1]+wr[6-yy+1];
       if (zw>=3) { sub1500(); continue; }
       if (fr[yy+1]==0 && fr[6-yy+1]==0) zw=0;
       sub1500();
     }
    zw=wr[5]; if (zw<(4-move)) zw=0;
    sub1500();
  }
```

Listing

```
sub1500()
{ if (zw==1) pu++;
  if (zw==2) pu+=10;
  if (zw>=3) pu+=100;
}
computer()
{ g_title(window,"Jetzt bin ich an der Reihe!");
  get_time(); start=time; tt=0; player= -1; bz=0; br[1]= -999;
  if (zu<2)
   { do
     { date(dummy,&datevec); bz=3+(datevec[6] % 3); }

     while(ar[54+bz]!=0);
   }
   else rekursion();
  zuege[zu]=bz; zu++;
  get_time(); time_atari+=1+(time-start); player=1; show_time();
}
rekursion()
{ tt++; player*= -1; xr[tt]=1;
  label1:
  if (ar[xr[tt]+9]!=0) goto label2;
  index=xr[tt]; depth(); yr[tt]=move; points(); pr[tt]=pu;
  br[tt+1]=0; pr[tt+1]=0;
  if (tt+1<=tmax) br[tt+1]= -999;
  if (tt==1) { if (pu>=100) { bz=xr[1]; return; } }
  ar[yr[tt]*9+xr[tt]]=player;
  if (pu>=100) { br[tt]=pu; ar[yr[tt]*9+xr[tt]]=0; xr[tt]=7; goto label2; }
  if (tt<tmax) rekursion();
  if (br[tt+1]== -999) br[tt+1]=0;
  if(pr[tt]-br[tt+1]>br[tt])
   { br[tt]=pr[tt]-br[tt+1]; if (tt==1) bz=xr[1]; }
  if (pr[tt]-br[tt+1]==br[tt])
   { date(dummy,&datevec);
     if (datevec[6]%2==0)
      { br[tt]=pr[tt]-br[tt+1]; if (tt==1) bz=xr[1]; } }
  ar[yr[tt]*9+xr[tt]]=0;
  label2:
  if (xr[tt]<7) { xr[tt]++; goto label1; }
  tt--; player*= -1;
}
print_items()
{ dummy=vsf_color(handle1,1); dummy=vsf_interior(handle1,2);
  dummy=vsf_style(handle1,4);
  corners[0]=12; corners[2]=corners[0]+104;
  for (y=1; y<6; y++)
    { corners[1]=220+y*30-16; corners[3]=220+y*30+5;
      v_bar(handle1,corners); }
  corners[0]=520; corners[2]=corners[0]+71;
  for (y=0; y<3; y++)
   { corners[1]=174+y*60; corners[3]=corners[1]+51;
     v_bar(handle1,corners); }
  v_gtext(handle1,36,370,"ABBRUCH");
  v_gtext(handle1,20,250,"Neues Spiel");
  v_gtext(handle1,16,310,"Spielstufe +");
  v_gtext(handle1,16,340,"Spielstufe -");
  v_gtext(handle1,16,280,"Zugrcknahme");
  v_gtext(handle1,528,253,"Spieler");
  v_gtext(handle1,524,313,"Computer");
  v_gtext(handle1,528,207,"Stufe ");
}
get_time()
{ date(dummy,&datevec);
  time=datevec[6]+datevec[5]*60+datevec[4]*3600;
}
show_time()
{ if (player<0)
   { y=274; time=time_player; } else { y=334; time=time_atari; }
  timebuf=time/3600; time-=timebuf*3600; itoa(timebuf,&buf1);
  if (timebuf<10) { strcat(&buf1,"0"); reverse(&buf1); } strcpy(&buf,buf1);
  strcat(&buf,"."); timebuf=time/60; time-=timebuf*60; itoa(timebuf,&buf1);
  if (timebuf<10) { strcat(&buf1,"0"); reverse(&buf1); } strcat(&buf,&buf1);
  strcat(&buf,":"); itoa(time,&buf1);
  if (time<10) { strcat(&buf1,"0"); reverse(&buf1); } strcat(&buf,buf1);
  v_gtext(handle1,524,y,buf);
}
```

```
print_max()
{ dummy=tmax-1; itoa(dummy,&buf); v_gtext(handle1,576,207,buf);
  do { vq_mouse(handle2,&buttons,&xpos,&ypos); } while(buttons!=0);
}
test()
{ leer=0; for(index=1; index<8; index++)
        { depth(); if (move>0) leer++; }
}
show_date()
{ date(dummy,&datevec);
  switch(datevec[3])
   { case 0: strcpy(&datum,"Sonntag"); break;
     case 1: strcpy(&datum,"Montag"); break;
     case 2: strcpy(&datum,"Dienstag"); break;
     case 3: strcpy(&datum,"Mittwoch"); break;
     case 4: strcpy(&datum,"Donnerstag"); break;
     case 5: strcpy(&datum,"Freitag"); break;
     case 6: strcpy(&datum,"Samstag"); break;
   }
  strcat(&datum,", der ");
  dummy=datevec[2]; itoa(dummy,&buf); strcat(&datum,buf); strcat(&datum,".");
  dummy=datevec[1]; itoa(dummy,&buf); strcat(&datum,buf); strcat(&datum,".19");
  dummy=datevec[0]; itoa(dummy,&buf); strcat(&datum,buf); strcat(&datum,", ");
  dummy=datevec[4]; itoa(dummy,&buf);
  if (dummy<10) { strcat(&buf,"0"); reverse(&buf); } strcat(&datum,buf);
  strcat(&datum,":"); dummy=datevec[5]; itoa(dummy,&buf);
  if (dummy<10) { strcat(&buf,"0"); reverse(&buf); } strcat(&datum,buf);
  strcat(&datum," Uhr");
  while(strlen(datum)<37) { strcat(&datum," "); }
  v_gtext(handle1,340,15,datum);
}
 abbruch()
 { v_hide_c(handle2); g_info(window,"Sie wollen wirklich aufhren? (j/n)");
   key=evnt_keybd(); key=key&255; if (key==74 || key==106) myexit=2;
   g_info(window," "); v_show_c(handle2,0);
 }
```

1000 Berlin

Computare oHG
☐ Keithstraße 18, 1000 Berlin 30
☐ Behaimstr. 3, 1000 Berlin 10
☎ (030) 2 13 90 21
Telex: 186 346 vom d



alpha computers g.m.b.h.
u. a. alphatronic, atari, commodore, dai, epson, sord mit pips, nec hard-/software nach maß — servicetechnik
Kurfürstendamm 121a, 1000 Berlin 31 (Halensee)
Telefon 030/8911082



Digital-Computer
Knesebeckstr. 76 · 1000 Berlin 12
Telefon
030-8827791



Vertragshändler
UNION ZEISS
Kurfürstendamm 57 • 1000 Berlin 15
Telefon 32 30 61

2000 Norderstedt



2080 Pinneberg

BPO
Ges. f. Beratung, Planung + Org. GmbH
Dingstätte 34
2080 Pinneberg
Telefon 0 41 01 - 2 60 71/72

2120 Lüneburg

Sienknecht
Bürokommunikation
Beratung - Verkauf - Werkstatt

Heiligengeiststr. 20, 2120 Lüneburg
Tel. 0 41 31 / 4 61 22, Btx 40 24 22
Mo.-Fr. 9⁰⁰-18⁰⁰ und Sa. 9⁰⁰-13⁰⁰

2160 Stade



Büromaschinen · EDV-Systeme
Neue Straße 5, 2160 Stade
Telefon: (04141) 23 64 + 23 84

2210 Itzehoe

Der Computerladen
Inhaber Ulrich Böbel Mathias Kentrup

Holzkamp 12 · 2210 Itzehoe
Telefon 0 48 21 / 33 90

2300 Kiel



Die Welt der Computer
Dreiecksplatz Nr. 7
2300 Kiel 1 · ☎ 04 31 / 56 70 42

2390 Flensburg

E C L
elektronic computer laden ohg

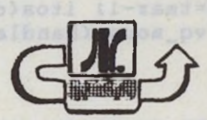
Norderstr. 94-96 · 2390 Flensburg
Telefon (04 61) 2 81 81 / 2 81 93

2800 Bremen

PS-DATA
Doventorsteinweg 41
2800 Bremen
Telefon 04 21 - 17 05 77

2850 Bremerhaven

HEIM- UND PERSONALCOMPUTER



Kurt Neumann
Georgstraße 71
2850 Bremerhaven
Tel. 04 71 / 30 21 29
HARDWARE · SOFTWARE · PAPIERWARE

2940 Wilhelmshaven

Radio Tiemann GmbH & Co. KG

2940 Wilhelmshaven
Telefon 0 44 21/2 61 45

2950 Leer



- HARDWARE-SOFTWARE
- SYSTEM-ENTWICKLUNG
- ORGANISATION
- EDV-SCHULUNG
- EDV-BERATUNG
- SERVICE-WARTUNG

Augustenstraße 3 · 2950 Leer
Telefon 04 91 - 45 89

3000 Hannover

COM DATA

Am Schiffgraben 19 · 3000 Hannover 1
Telefon 05 11 - 32 67 36



DATALOGIC
COMPUTERSYSTEME
ATARI ST- COMPUTER HARDWARE SOFTWARE
BERATUNG SERVICE VERKAUF
CALENBERGER STR. 26
3000 HANNOVER 1
TEL.: 0511 - 32 64 89

3040 Soltau

F & T Computervertrieb

Am Hornberg 1
(Industriegeb. Almhöhe)
3040 Soltau
Tel. 0 51 91 / 165 22

3100 Celle

Ludwig Haupt jr.
Büro-Einkaufs-Zentrum
Gerhard-Kamm-Straße 2
Ruf 8 30 45, Postfach 140
3100 Celle

3170 Gifhorn

COMPUTER-HAUS GIFHORN
Braunschweigerstr. 50
3170 Gifhorn
Telefon 0 53 71 - 5 44 98

3300 Braunschweig

COMPUTER STUDIO BRAUNSCHWEIG
Rebenring 49-50
3300 Braunschweig
Tel. (05 31) 33 32 77/78

3320 Salzgitter



Beratung und Vertrieb
für Computer und Bildschirmtext
Berliner Str. 54 · 3320 Salzgitter 1
Telefon (0 53 41) 4 40 91-2

3400 Göttingen

Büroeinrichtungs-Zentrum


3400 Göttingen-Weende
Wagenstieg 14 — Tel. 0551/34031

3500 Kassel

Hermann Fischer GmbH
autorisierter ATARI-Fachhändler
Rudolf-Schwander-Str. 5, 9 + 13
3500 Kassel
Tel. (05 61) 70 00 00

3550 Marburg

L W M
COMPUTER SERVICE
Biegenstraße 43
3550 Marburg/Lahn
06421-622236

4000 Düsseldorf

BERNSHAUS GmbH
Bürotechnik — Bürobedarf
Cäcilienstraße 2
4000 Düsseldorf 13 (Benrath)
Telefon 02 11 - 71 91 81

HOCO
EDV ANLAGEN GMBH
Flügelstr. 47
4000 Düsseldorf
Tel. 02 11 - 77 62 70

4050 Mönchengladbach



Hindenburgstr. 249
4050 Mönchengladbach
Tel. 0 21 61 - 187 64

4190 Kleve

Computer Feldmann + Luft
4190 Kleve-Kellen Emmericher Str. 223
Telefon 0 28 21/9 10 38 · Telex 811 797

4200 Oberhausen

KAMP
Büro- und Computersysteme
Vestische Straße 89/91
4200 Oberhausen 12 (Osterfeld)
Fernruf (02 08) 89 00 86
Fernschreiber 8 56 578

4350 Recklinghausen

COMPUTER CENTRALE
Douastr. 1 · 4350 Recklinghausen
Telefon 0 23 61 - 4 57 08

4400 Münster



COMPUTER SYSTEME GMBH
Daimlerweg 39 · 4400 Münster
Telefon 02 51 / 71 99 75 - 9

4420 Coesfeld



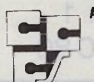
4422 Ahaus

ATARI · Apricot · Epson · Fuejitsu · Molecular
Computer-Systeme + Software
OCB
Wallstraße 3 · 4422 Ahaus
Telefon 0 25 61/50 21

4500 Osnabrück

Heinicke-Electronic
Kommenderiestr. 120 · 4500 Osnabrück
Telefon 05 41 - 8 27 99
Wir liefern Micro-Computer seit 1978

4600 Dortmund



Atari, Genie, Schneider, Tandy, Brother, Star, Memorex, BASF, Verbatim
cc Computer Studio GmbH
Software-Hardware-Beratung
Service-Eilversand
Ihre Ansprechpartner:
v. Schablinski
Jan P. Schneider
Elisabethstraße 5
4600 Dortmund 1
T. 0231/528184 · Tx 822631 cccsd

City Elektronic
Güntherstraße 75
4600 Dortmund
Telefon 02 31/57 22 84

5900 Siegen

6000 Frankfurt

6100 Darmstadt

6200 Wiesbaden

6240 Königstein

6300 Gießen

6330 Wetzlar

6350 Bad Nauheim

6400 Fulda

6457 Maintal

6500 Mainz

6520 Worms

6600 Saarbrücken

6700 Ludwigshafen

6720 Speyer

MKV Computermarkt
Gilgenstraße 4
6720 Speyer
Telefon 06232-77216

6730 Neustadt

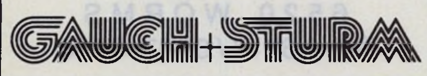
Felten & Meier Computersysteme
Exterstr. 4 · 6730 Neustadt
Tel. 06321/88994

6750 Kaiserslautern

C.O.S.
COMPUTER ORG. GmbH
Karl-Marx-Straße 8
6750 Kaiserslautern
Telefon (0631) 65061-62

6800 Mannheim

Computer Center
Am Hauptbahnhof GmbH
L 14, 16-17
6800 Mannheim
Tel. 0621/20983-4



Computersysteme + Textsysteme
6800 Mannheim 24
Casterfeldstraße 74-76
☎ (0621) 850040 · Teletex 6211912

6900 Heidelberg

Heidelberger Computer-Center
Bahnhofstraße 1
6900 Heidelberg
Telefon 06221/27132

JACOM COMPUTERWELT
Hardware · Software
Schulung · Service
Mönchhofstraße 3 · 6900 Heidelberg
Telefon 06221/410514-550

7000 Stuttgart

BNT Computerfachhandel Seibel & Co. oHG
Der kleine mit der **großen** Leistung
Markstraße 48 · 7000 Stuttgart 50
Telefon 0711/558383

7022 L.-Echterdingen

Autorisierter ATARI-System-Fachhändler für **520 ST** 130 XE
ATARI


Michael Matrai
Bernhauser Str. 8
7022 L.-Echterdingen
☎ (0711) 797049

7030 Böblingen

MCA Computer-Center
Sindelfinger Allee 1
7030 Böblingen
Tel. 07031/223618

7100 Heilbronn

Unser Wissen ist Ihr Vorteil
Walliser & Co.
Mönchseestraße 99
7100 Heilbronn
Telefon 07131/60048

7150 Backnang

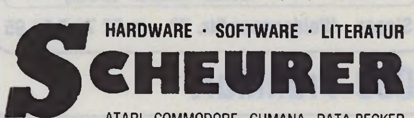


7450 Hechingen



Gesellschaft für Datenverarbeitung mbH
Computer · Drucker
Zubehör · Fachliteratur
Schloßplatz 3 · 7450 Hechingen
Telefon 07471/14507

7475 Meßstetten

Ihr ATARI-Systemhändler im Zollern-Alb-Kreis
HEIM + PC-COMPUTERMARKT


7475 Meßstetten 1 · Hauptstraße 10 · 07431/61280

7480 Sigmaringen



7500 Karlsruhe

MKV Computermarkt
Rüppurer Straße 2d
7500 Karlsruhe
Telefon 0721-373071

papierhaus erhardt
Am Ludwigsplatz · 7500 Karlsruhe
Tel. 0721-23925

7600 Offenburg

FRANK LEONHARDT ELECTRONIC
Ihr Fachgeschäft für Microcomputer · Hifi · Funk
In der Jeuch 3
7600 Offenburg
Telefon 0781/57974

7700 Singen

U. MEIER
Groß- und Einzelhandel
7700 Singen-Htwl.
Postfach 447

7730 VS-Schwenningen

BUS BRAUCH & SAUTER COMPUTER TECHNIK
Villinger Straße 85
7730 VS-Schwenningen
Telefon 0 77 20 / 3 80 71-72

7750 Konstanz

computertechnik
rösler
D-7750 Konstanz
Zasiusstr. 35 · ☎ 0 75 31 / 2 18 32

7800 Freiburg

CDS
EDV-Service GmbH
Windausstraße 2
7800 Freiburg

computer aktuell
Südbadens kompetenter Computer-Partner.
Kaiser-Joseph-Str. 232
7800 Freiburg, Tel.: 07 61 / 2180 225

7890 Waldshut-Tiengen

hettler-data
service gmbh
Lenzburger Straße 4
7890 Waldshut-Tiengen
Telefon 0 77 51 / 30 94

7900 Ulm

HARD AND SOFT COMPUTER GMBH
Ulms großes Fachgeschäft
für BTX, Heim- u. Personalcomputer
Herrenkellergasse 16 · 7900 Ulm/Donau
Telefon 07 31 / 26 99

COMPUTERSTUDIO
Büro & Datentechnik
Büro & Datentechnik
Claus Wecker
Hafenbad 18/1 + Frauenstr. 28
7900 Ulm/Do.
Telefon (07 31) 6 80 76l

8000 München



COMPUTER + BÜROTECHNIK
COMPUTER · SOFTWARE · PERIPHERIE
BERATUNG · TECHN. KUNDENDIENST
INGOLSTÄDTER STR. 62 L
EURO-INDUSTRIE-PARK · 8000 MÜNCHEN 45
TELEFON 089 / 311 30 66 · TELETEX 89 83 41

MSG
Marketing u. Service
Autorisiertes Atari-Service Center
Adelmannstr. 5
8000 München 82
Telefon 0 89 / 4 30 03 33

8032 Gräfelfing

Pro-CE
8032 Gräfelfing
Tel. 0 89 / 8 54 54 64

8070 Ingolstadt

DREYER GMBH
Elektrotechnik
Manchinger Straße 125
8070 Ingolstadt
Tel. 08 41 / 65 90

8170 Bad Tölz

Elektronik Center Bad Tölz
Wachterstraße 3
8170 Bad Tölz
Telefon 0 80 41 / 4 15 65

8200 Rosenheim

COMPUTING
Theodor-Gietl-Str. 3 · 8200 Rosenheim · Tel.: 0 80 31 / 6 80 21

8220 Traunstein

computer studio cs
BÜROMASCHINEN
Ludwigstraße 3 · 8220 Traunstein
Stadtplatz 10 · Tel. 0861-14767 o. 3905

8263 Burghausen

JASKULSKI
Hard- u. Software
Mautnerstr. - 8263 Burghausen
Telefon 0 86 77 / 6 33 20

8300 Landshut

BÜRO-DALLMER
Altstadt 69
8300 Landshut
Telefon 08 71 / 2 10 62 - 64

8400 Regensburg

C-SOFT GMBH
Programmentwicklung & Hardware
Holzfällerstraße 4
8400 Regensburg
Telefon 09 41 / 8 39 86

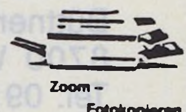
Zimmermann elektroland
8400 Regensburg · Dr.-Gessler-Str. 8 · ☎ 09 41 / 9 50 85
8390 Passau · Meraner-Str. 5 · ☎ 08 51 / 5 10 16

8490 Cham

AGP-SHOP®
INH. A. STEUER
Auf der Schanze 4 · 8490 Cham/Opf. · Telefon (0 99 71) 97 23

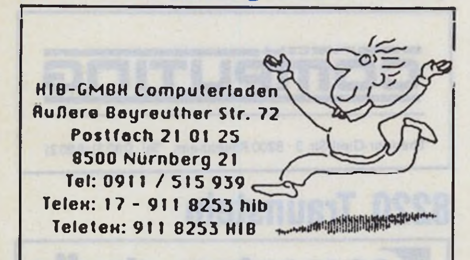


Ihr Fachgeschäft für Microcomputer und Schreibwaren



Zoom - Fotokopieren

8500 Nürnberg



8520 Erlangen



Dresdener Str. 5 · Friedrichstr. 9
8520 Erlangen · Telefon 12 01 - 0

Computerservice Decker
Meisenweg 29 - 8520 Erlangen
Telefon 0 91 31 / 4 20 76

8590 Marktredwitz

STEINWALD ELEKTRONIK GMBH
Am Steingrund 1 · 8590 Marktredwitz
Telefon 0 92 31 / 6 20 18

8600 Bamberg



8670 Hof

COMPUTER-CENTER-BURGER
Spezialist für Personal- und Home-Computer, Programme
Zubehör, Beratung, Service
8670 Hof · Leimitzer Straße 11 · Telefon 09281/40075 Abt. Computer



8700 Würzburg

HALLER GMBH
Fachgeschäft für Mikrocomputer
Büttnerstraße 29
8700 Würzburg
Tel. 09 31 / 1 67 05

8700 Würzburg



8720 Schweinfurt

Uhlenhuth GmbH
Computer + Unterhaltungselektronik
Albrecht-Dürer-Platz 2
8720 Schweinfurt
Telefon 0 97 21 / 65 21 54

8752 Mömbris



DIE EXPERTEN FUR MICROCOMPTER
Im Kahltal Zentrum, 8752 Mombris
Telefon: (0 60 29) 65 20 oder 14 10

ATARI 520 ST
APRICOT
IBM Komp
Festplatten

Hardware
Software
Schulung

8850 Donauwörth



☎ 09 06 - 60 99
Altes Sträßle 28
8850 Donauwörth
Ihr Büroausstatter

8900 Augsburg

Vertragshändler für IBM Personal-Computer

PROFI COMPUTER STUDIO

KARSTADT Aktiengesellschaft
Bürgermeister-Fischer-Str. 6-10
8900 Augsburg
Telefon(08 21)31 53-4 16

Adolf & Schmoll Computer Studio
Hörbrotstr. 6 · 8900 Augsburg
Telefon (08 21) 52 85 33
Wir sind außerdem autorisierte Service-Fachwerkstatt für:
Schneider ATARI
commodore

8901 Augsburg-Vogelsang

VIDEO + COMPUTER HANDELS GMBH
Steppacher Straße 8 A
8901 Augsburg-Vogelsang
Telefon 08 21 / 48 20 76

8910 Landsberg

Szeredy Elektronik GmbH
Computervertrieb
Sandauerstr. 253
8910 Landsberg · ✆ 0 81 91 / 3 95 00

8940 Memmingen

EDV-Organisation
Hard- + Software
Manfred Schweizer KG
Benninger Str. 34, Tel. 0 83 31 / 1 22 20
8940 Memmingen

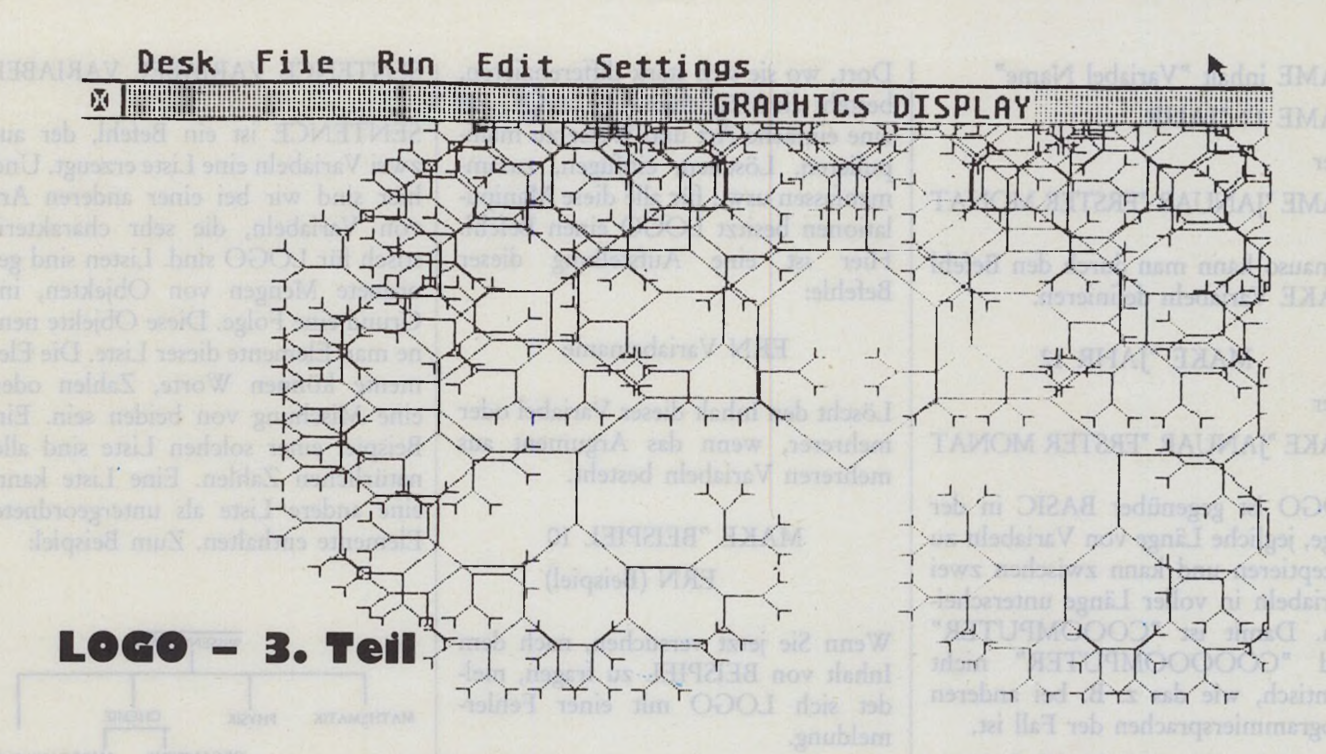
Österreich

W + H.
Computerhandel Ges m.b.H.
Förstergasse 6/3/2 · 1020 Wien
Tel. 02 22 - 3 38 45 92

Zupan GmbH
Mandellstr. 23
8010 Graz
Telefon 03 16 - 70 28 40
78 39 23

Schweiz

PAUS-Electronic
Ihr Computer-Fachhandel
Industriestr. 30
Pestalozzistr. 11
CH-9400 Rorschach ✆ 071-41 18 85



LOGO – 3. Teil

– oder eine Einführung in Turtle-Geometrie

In der letzten Folge unseres LOGO-Kurses haben wir die elementaren Graphik-Befehle sowie mathematische und logische Funktionen kennengelernt. Wir haben auch Variabeln verwendet, sind dann aber nicht weiter auf sie eingegangen.

In dieser Folge werden wir uns, unter anderem, mit den Möglichkeiten der Verwaltung von Variabeln bei LOGO beschäftigen.

GLOBALE, LOKALE VARIABELN UND LISTEN

Wenn wir den ganzen Speicherbereich unseres Computers mit einem riesigen Schrank voller Schubladen vergleichen, die aufwärts durchnummeriert sind, dann ist eine Variabel nichts anderes als eine Schublade und deren Inhalt der momentane Wert, den wir ihr zugewiesen haben. Wir kennen von unserer allerersten Prozedur eine Variabelart, die wir Parameter genannt haben.

```
TO DREIECK :SEITE
   REPEAT 3 (FD :SEITE RT 120)
END
```

In diesem Beispiel ist SEITE ein Parameter, das veschiedene Werte annehmen kann. Dieser Wert beschränkt sich nicht auf diese Prozedur, sondern kann auch auf eine andere übertragen werden. Wir stellen eine andere Prozedur her:

```
TO QUADRAT :SEITE
   repeat 4 (FD :SEITE RT 90)
END
```

Und jetzt tippen Sie ein:

```
TO HAUPT :SEITE
   DREIECK :SEITE
   QUADRAT :SEITE
END
```

In Direkt-Modus geben Sie ein:

```
HAUPT 50
```

Wie Sie sehen, dient das gleiche Parameter für beide Programme. Auf diese Art und Weise Zahlen einzugeben, ist durchaus komfortabel und sicher eines der wichtigsten Merkmale von LOGO. Natürlich kennt unsere Igel-Sprache andere Arten von Variabeln, und im Vergleich dazu sind andere Sprchen sehr arm dran. Wir kennen von BASIC einen anderen Weg, eine Variabel zu definieren:

```
A = 15
A$ = "ST-COMPUTER"
```

Das erste Beispiel stellt eine numerische Variabel dar und die zweite eine sogenannte STRIGS Variabel. Je nach den verschiedenen BASIC-Dialekten wird die Länge eines „Variabel-Namens" bestimmt. In LOGO geschieht dies alles etwas anders.

LOGO besitzt zur Definition einfacher, globaler Variabeln zwei Arten, oder besser gesagt zwei verschiedene Befehle:

NAME inhalt "Variabel Name"

NAME 12 "JAHR

oder

NAME "JANUAR "ERSTER MONAT

Genauso kann man durch den Befehl MAKE Variabeln definieren.

MAKE "JAHR 12

oder

MAKE "JANUAR "ERSTER MONAT

LOGO ist gegenüber BASIC in der Lage, jegliche Länge von Variabeln zu akzeptieren und kann zwischen zwei Variabeln in voller Länge unterscheiden. Damit ist "COOOMPUTER" und "COOOOOMPUTER" nicht identisch, wie das z. B. bei anderen Programmiersprachen der Fall ist.

Genauso, wie eine Variabel einen Wert zugewiesen bekommen kann, ist auch die Möglichkeit gegeben, daß sie ihn selbst ausrechnet. In dem nächsten Beispiel wird eine Variabel ZAHL immer inkrementiert und angezeigt, bis sie ihren Endwert (der von dem Anwender eingegeben wird) erreicht hat.

```
TO ZAHL
CT
PRINT [BITTE EINE BELIEBIGE ZAHL (GROESSER NULL) EINGEBEN]
MAKE "ZAHL RQ
MAKE "L 0
LABEL "SCHLEIFE
PRINT :L
MAKE "L :L + 1
IF :L < :ZAHL [GO "SCHLEIFE]
END
```

Zum Programm ist nicht viel zu sagen. Zuerst wird eine Zahl angefordert, die der Variabel ZAHL zugewiesen wird. Dann wird die Lauf-Variabel „L" mit dem Wert 0 initialisiert. In der übernächsten Zeile wird der laufende Wert von „L" angezeigt und dann in Eins (1) inkrementiert. Wenn „L" den Wert von ZAHL noch nicht erreicht hat, wird auf die Zeile, die mit dem Label SCHLEIFE versehen ist, zurückgekehrt. Der gleiche Vorgang wiederholt sich, bis der Endwert erreicht ist.

Wie Sie sicher bemerkt haben, weicht die Handhabung von numerischen Variabeln in LOGO nicht so gravierend ab, wie das in BASIC der Fall wäre. Dort, wo sie sich stark differenzieren, besteht die Möglichkeit, in LOGO auf eine einfache Art und Weise zu manipulieren. Löschen, einfügen, zusammenfassen usw.: für alle diese Manipulationen besitzt LOGO einen Befehl. Hier ist eine Aufstellung dieser Befehle:

ERN Variabelname

Löscht den Inhalt dieser Variabel oder mehrerer, wenn das Argument aus mehreren Variabeln besteht.

MAKE "BEISPIEL 10
ERN (Beispiel)

Wenn Sie jetzt versuchen, nach dem Inhalt von BEISPIEL zu fragen, meldet sich LOGO mit einer Fehlermeldung.

ERALL

Löscht alle Variabeln. Wenn man die Wirkung dieser Befehle nur auf eine bestimmte Prozedur beschränken möchte, muß der Name dieser Prozedur als Argument versehen werden.

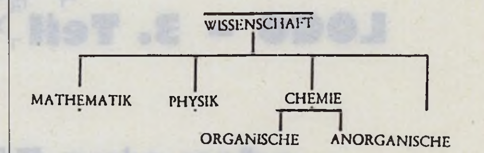
ERALL "TEST

Löscht alle Variabeln der TEST-Prozedur



SENTENCE VARIABEL VARIABEL

SENTENCE ist ein Befehl, der aus zwei Variabeln eine Liste erzeugt. Und hier sind wir bei einer anderen Art von Variabeln, die sehr charakteristisch für LOGO sind. Listen sind geordnete Mengen von Objekten, im Grund eine Folge. Diese Objekte nenne man Elemente dieser Liste. Die Elemente können Worte, Zahlen oder eine Mischung von beiden sein. Ein Beispiel einer solchen Liste sind alle natürlichen Zahlen. Eine Liste kann eine andere Liste als untergeordnete Elemente enthalten. Zum Beispiel:



Als Liste sieht dies so aus:
WISSENSCHAFT (MATHEMATIK PHYSIK CHEMIE (ORGANISCHE ANORGANISCHE)...)

So eine Liste nennt man Liste zweiter Ordnung. Würde die untergeordnete Liste noch eine Liste beinhalten, wäre das eine Liste dritter Ordnung.

Bei der Datenverarbeitung ist diese Möglichkeit strukturierter Elemente von großer Bedeutung. Denken Sie an eine Datenverwaltung, in der unter einem Namen eine Menge von „untergeordneten" Elementen stecken können. Eine Liste wird nicht nur in LOGO durch SENTENCE erzeugt, sondern

auch durch den Befehl MAKE, wobei die Befehle innerhalb eckiger Klammern 'stehen müssen

MAKE "TIERE (HAUSTIERE (HUND KATZE) WILDTIERE (LOEWE WOLF)

Mit Hilfe eines graphischen Beispiels werden wir die Wirkung von SENTENCE verdeutlichen. Das setzt voraus, daß wir einiges mehr über den Aufbau des Graphik-Windows wissen müssen. Der Graphik-Bildschirm bei LOGO besteht bei einem Schwarz-Weiß-Monitor aus 640 Pixels in der Horizontalen und 400 in der Vertikalen. Die Anfangskoordinate befindet sich in der Mitte und beträgt die 0,0.

Rechts von dem Nullpunkt wachsen die Schritte ins Positive, links davon ins Negative. Ebenso verläuft dies in die obige Richtung und umgekehrt nach unten.

Um die Schildkröte in die X-Achse zu verlagern, geben Sie ein:

SETX -300

Jetzt steht das Zeichensymbol links auf dem Bilschirm

SETY 150

Versetzt unseren Freund um 150 Schritte nach oben.

SETPOS X Y

ist eine Zusammenfassung beider obengenannten Positionierungsbefehle.

Alle drei Befehle sind nicht nur im Direkt-Modus erreichbar, sondern werden auch in Programme eingebaut. Damit ist die Richtung der Schildkröte ständig einfach zu finden. Solche Probleme treten bei der Darstellung von Funktionen immer wieder auf. Durch Eingabe veränderlicher Variabeln erreichen wir eine Steuerung in alle möglichen Richtungen. Verdeutlichen wir es uns zuerst an einem einfachen Beispiel:

```
TO POSITIONIERUNG
MAKE " X 0
MAKE "Y 0
LABEL "LOOP
SETX :X
SETY :Y
MAKE "X :X + 1
MAKE "Y :Y + .5
IF :Y 100 (GO "LOOP)
END
```

Die Schildkröte wird erst durch beide Befehle SETX,SETY auf den Punkt 0 gesetzt. Dann werden beide Werte inkrementiert, solange Y nicht den Wert 100 erreicht hat. Durch die Schleife wird das Symbol immer auf eine neue Stelle positioniert. Natürlich werden Sie sich überlegen, warum man den Befehl SETPOS nicht benutzt. Das hätte das ganze nur erleichtert.

Ersetzen Sie die beiden SET-Befehle durch

SETPOS :X :Y

Jetzt lassen Sie das Programm neu laufen. LOGO erkennt beide Parameter nicht und meldet sich mit einer Fehlermeldung.

Damit nicht alles im Gebiet der „grauen“ Theorie bleibt, werden wir ein Programm eintippen, das die Darstellung einer Sinuskurve ermöglicht, die aus lauter Ellipsen besteht. Ebenfalls werden wir sehen, daß sich durch kleine Änderungen auch andere Formen von Kurven bilden lassen.

Das Programm besteht aus vier kleinen Routinen. POSITION setzt die Schildkröte an den Ausgangspunkt, der sich ganz links auf dem Bildschirm befindet. Dann werden einige Variabeln definiert

"M" ist ein Faktor, der für die Schritte auf der X-Achse verantwortlich ist
"F2" ist der Wert für die Sinusberechnung
"CX" ist der x-Radius der Ellipse
"CY" ist der y-Radius der Ellipse

Der Befehl MAKE "CC SE :CX :CY erzeugt, wie bekannt, aus zwei Variabeln eine Liste mit dem Namen CC, deren Elemente CX und CY sind. Die zweite Prozedur errechnet für Y den Sinuswert, der den Verlauf der Kurve beeinflussen wird. X wird gradlinig berechnet und strebt von dem Punkt -300 zu +300. LAUF ist die Variabel, die die zwei errechneten Werte für den Anfang X sowie Y umfaßt. Wie Sie sehen, handelt es sich auch hier um eine Liste mit den folgenden Variabeln (XY).

PLOT ist die Routine, die das Zeichen bewerkstelligt. Zuerst wird die Schildkröte gemäß der Variabel LAUF auf die Position gesetzt, von der aus sie zu zeichnen beginnt. Dann wird CP definiert. CP ist eine Liste zweiter Ordnung mit folgenden Elementen:

CP (LAUF (X Y) CC (CX CY))

Das ist in diesem Falle nötig, weil die Ellipse vier ARgumente braucht.

Ellipse X Koordinate Y Koordinate X Radius Y Radius

Die HAUPT-Routine ist die Prozedur, mit der das Programm aufgerufen wird. Am Anfang wird POSITION ausgeführt, dann LABEL für die nötige Schleife angebracht. Die beiden Routinen RECHNUNG und PLOT werden solange ausgeführt, bis T den Wert von 55 erreicht hat.

Jetzt vergrößern Sie den Graphik-Bildschirm und geben im Direkt-Modus ein:

HAUPT

Die Größe der einzelnen Ellipsen läßt sich durch Änderungen von CX und CY bestimmen, die Dichte der Kurve durch Änderung der Schrittweite bei „T“. Anstelle von Ellipsen besteht

auch die Möglichkeit, die Kurve durch Kreise darzustellen. Ersetzen Sie bei der Prozedur PLOT den Befehl ELLIPSE durch CIRCLE und lassen Sie das Programm erneut laufen. Um eine einfache, aus Punkten bestehende Sinuskurve zu zeichnen, entfernen Sie die beiden Variabeln CX und CY aus der Prozedur POSITION.

PLOT muß nun wie folgt aussehen:

```
TO PLOT
SETPOS : Lauf
END
```

Wenn Sie jetzt das Programm laufen lassen, werden Sie eine einfache Sinuskurve sehen, die sich von der linken Seite des Bildschirms zur rechten erstreckt.

Eindrucksvoller ist unser Programm durch die Darstellung sogenannter Lisajour-Figuren zu gestalten. Damit sind einige Änderungen notwendig. Die Prozedur POSITION wird jetzt so aussehen:

```
TO POSITION
CS
PU
SETX -100
FD
MAKE "M 20
MAKE "F2 4
MAKE "F1 5
MAKE "T 0
END
```

RECHNUNG muß auch einige Änderungen in Kauf nehmen.

```
TO RECHNUNG
MAKE "Y 50 * SIN (:F2 * :T)
MAKE "XX 80 * COS (:F1 * :T)
MAKE "X -100 + :XX
MAKE "RY 20 - :Y
MAKE "LAUF SE :X :RY
MAKE "T :T + 1
END
```

Die PLOT-Routine bleibt, wie bei der einfachen Darstellung der obengenannten Sinuskurve. Die Variabl „T" wird in dem Programm HAUPT mit der Zahl 500 verglichen.

Andere Kurven lassen sich genauso einfach darstellen. Mit ein wenig Phantasie kann man diese Programme in eine universelle Plotter-Routine für mathematische Funktionen gestalten.

```
TO POSITION
CS
PU
SETX -300
PD
MAKE "M 10
MAKE "F2 20
MAKE "T 1
MAKE "CX 40
MAKE "CY 20
MAKE "CC SE :CX :CY
END

TO RECHNUNG
MAKE "Y 150 * SIN (:F2 * :T)
MAKE "X :T * :M
MAKE "X -300 + :X
MAKE "RY :Y
MAKE "LAUF SE :X :RY
MAKE "T :T + 0.15
END

TO PLOT
PU
SETPOS :LAUF
MAKE "CP SE :LAUF :CC
PD
ELLIPSE :CP
END

TO HAUPT
POSITION
LABEL "HIER
RECHNUNG
PLOT
IF :T < 55 [GO "HIER]
END
```

Alle bisher genannten Variabeln haben eines gemeinsam: sie gehören zu dem Typ der Global-Variabeln. Einmal definiert, wirken sie in allen Prozeduren, in denen sie wieder auftauchen.

LOGO kennt einen anderen Typ von Variabeln, die nur auf bestimmte Prozeduren beschränkt sind; diese Variablart nennt man Lokalvariabeln.

LOCAL Variabelname

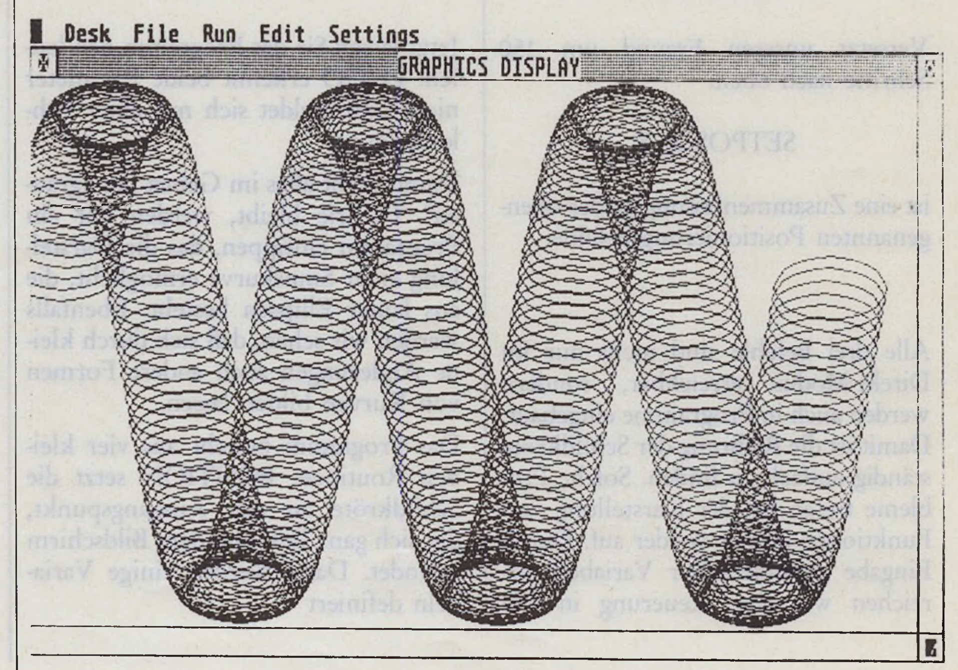
Wenn mehrere Variabeln als Local definiert werden, muß der Befehl in Klammern stehen:

(LOCAL Variabl Variabl Variabel...)

LOCAL "ST

MAKE "ST "REDAKTION

Loka-Variabeln werden wie normale Variabeln behandelt, nur mit der Einschränkung, daß sie sich auf die Prozedur beschränken, die sie beinhalten. Andererseits sind sie insofern wichtig, daß sie Platz sparen.



LOGO und die STRINGS

Wie schon früher erwähnt wurde, kann LOGO auch Strings verarbeiten, und das alles schneller als in BASIC. Durch eine Anzahl von Befehlen ist LOGO in der Lage Strings zu isolieren, teilen, tauschen usw.. Folgend eine Zusammenfassung der Befehle zur Stringverwaltung:

COUNT "COMPUTER ergibt 8
COUNT (1 3 5) ergibt 3

COUNT ermittelt die Anzahl von Buchstaben eines Wortes oder die Anzahl von Elementen einer Liste.

ITEM Position Wort oder Liste

Gibt die Buchstaben eines Wortes oder die Elemente einer Liste auf die Position zurück.

ITEM 3 "COMPUTER ergibt M

PIECE VON BIS Wort oder Liste

PIECE schneidet ein Wort oder eine Liste von dem ersten Argument bis zum zweiten

PIECE 1 3 "ATARI ergibt ATA

WORD Wort 1 Wort 2

Setzt zwei Worte zusammen.

WORD "ST- "COMPUTER ergibt ST-Computer

FIRST Wort,Liste

Gibt die ersten Buchstaben oder Elemente einer Liste zurück.

FIRST "ATARI ergibt A

FIRST (10 20 30) ergibt 10

LAST WORT,Liste

Gibt den letzten Buchstaben eines Wortes oder das letzte Element einer Liste zurück

LAST "ATARI ergibt I

LAST (10 20 30) ergibt 30

BUTFIRST Wort,Liste

Ergibt alle Buchstaben eines Wortes oder alle Elemente einer Liste bis auf den ersten wieder

BUTFIRST "COMPUTER ergibt OMPUTER

BUTLAST WORT,Liste

ist das Gegenstück von BUTFIRST. Es werden alle Buchstaben eines Wortes sowie alle Elemente einer Liste zurückgegeben, mit Ausnahme des letzten

FPUT Wort, Zahl oder Liste
Wort, Zahl oder Liste

Setzt zwei Ausdrücke zu einem zusammen, mit dem ersten Argument vorne.

FPUT (10 20 30) (40 50 60)
ergibt (10 20 30 40 50 60)

LPUT Wort,Zahl oder Liste
Wort, Zahl oder Liste

Setzt zwei Ausdrücke zu einem zusammen, mit dem ersten Argument hinter dem zweiten.

LPUT "RI "ATA ergibt ATARI

.REPLACE Position Liste oder Variabel Liste

.REPLACE ersetzt ab der angegebenen Position das zutreffende Element einer Liste durch einen Variabel-Inhalt oder eine Liste

MAKE "COMPUTER (HARDWARE SOFTWARE LEBENSMITTEL)
.REPLACE 3 :COMPUTER (LOGIK)

ergibt

?:COMPUTER
(HARDWARE SOFTWARE (LOGIK))

.REPTAIL Position Liste oder Variabel Liste

.REPTAIL fügt den Inhalt einer Liste, einer Variabel, aber der angegebenen Positon in eine zweite Liste und löscht die übrigen Elemente dieser zweiten

MAKE "ZAHL (1 2 3 4 8 9 0)
.REPTAIL 5 :ZAHL (5 6 7)

Ergibt

?:ZAHL
(1 2 3 4 5 6 7)

SORT (Liste)

LOGO kann ohne großen Aufwand Elemente einer Liste sortieren. Die Sortierung geschieht nach der Größe von ASCII Code in steigender Form

SORT (Z C W A) ergibt (A C W Z)

SHUFFLE (Liste)

Eine nach ASCII geordnete Liste wird wieder durcheinander gebracht.

EQUALP Wort, Zahl oder Liste
Wort, Zahl oder Liste

Ergibt TRUE, wenn die zwei zu vergleichenden Worte, Zahlen oder Listen gleich sind.

EQUALP "COMPUTER "COMPUTER
ergibt TRUE

dagegen

EQUALP "COMPUTER "COM
ergibt FALSE

Unser nächstes Beispiel veranschaulicht uns die Handhabung von Strings Variabeln sowie Listen mit Strings. Es handelt sich um ein einfaches Dateiprogramm, das jedoch einige Wünsche offen läßt. LOGO besitzt nicht viele Möglichkeiten, Daten auf einem externen Massenspeicher zu verwalten; Daten und Programme werden zusammen auf Diskette abgespeichert. Insofern sit diese Art mit einer sequentiellen Datei zu vergleichen. Das hat den Nachteil, daß man verschiedene Versionen des gleichen Programms auf der Diskette hat. Die einzige Lösung ist, ältere Versionen zu löschen.

Das Programm besteht aus fünf kleinen Routinen, die nacheinander nach Bedarf aufgerufen werden können. Die VOR-Routine initialisiert die Zähl-Variabeln und ruft das Subroutine-Menü auf. MENÜ bringt auf dem Bildschirm die verschiedenen Möglichkeiten, die dem Anwender zur Verfügung stehen und ermöglicht eine Wahl über die Tastatur. Je nach Auswahl werden die anderen Prozeduren aufgerufen. EINGEBEN ist verantwortlich für die Aufnahme von Daten. Die Suche und Ausgabe von Daten wird durch die Prozedur AUSGABE verrichtet. Erfaßte Daten werden durch die Unter-Routine ABSPEICHER auf die Diskette samt Programm gespeichert.

Damit ist unser LOGO-Kurs zu einem Ende gekommen. Wir hoffen, daß diese Folge eine Anregung ist, in dieser Sprache zu programmieren. Wir würden uns freuen, Ihre Meinung darüber zu erfahren. In Zukunft werden wir weitere kleine Programme veröffentlichen, die das Arbeiten in dieser Sprache verdeutlichen.

Viel Spaß beim Programmieren in LOGO.

(MM)

Desk File Run Edit Settings

LOGO DIALOGUE

* * * * * * ADRESS BUCH * * * * * *

- 1 - EINGABE
- 2 - AUSGABE
- 3 - ABSPEICHER
- 4 - END

BITTE WAEHLEN

```
TO MENU
CT
TYPE [* * * * * * ADRESS BUCH * * * * * *]
REPEAT 4 [PR "]
PR [- 1 - EINGABE]
PR [- 2 - AUSGABE]
PR [- 3 - ABSPEICHER]
PR [- 4 - END]
REPEAT 4 [PR "]
TYPE [BITTE WAEHLEN  ]
MAKE "WA FIRST RL
IF :WA = 1 [EINGEBEN]
IF :WA = 2 [AUSGABE]
IF :WA = 3 [ABSPEICHER]
IF :WA = 4 [STOP]
MENU
END

TO EINGEBEN
MAKE "ZL :ZL + 1
CT
PR [* * * * * * ADRESS BUCH * * * * * *]
REPEAT 4 [PR "]
TYPE [NAME ? ]
MAKE "NA WORD "NAME :ZL
MAKE "NAM RL
PPROP :NA "NAME :NAM
TYPE [ADRESSE ? ]
MAKE "AD WORD "ADRESS :ZL
MAKE "ADR RL
PPROP :AD "ADRESS :ADR
TYPE [TELEFON ? ]
MAKE "TE WORD "TELE :ZL
MAKE "TEL RL
PPROP :TE "TELE :TEL
END

TO AUSGABE
MAKE "Z2 0
CT
PR [* * * * * * ADRESS BUCH * * * * * *]
TYPE [BITTE GEBEN SIE EINER NAME EIN ]
MAKE "SN RL
LABEL "SCHLEIF
MAKE "Z2 :Z2 + 1
MAKE "NA WORD "NAME :Z2
IF GPROP :NA "NAME = :SN [GO "AUS]
IF :Z2 < 100 [GO "SCHLEIF]
GO "NG
LABEL "AUS
TYPE [NAME ] PR GPROP :NA "NAME
MAKE "NA WORD "ADRESS :Z2
TYPE [ADRRESSE ] PR GPROP :NA "ADRESS
MAKE "NA WORD "TELE :Z2
TYPE [TELEFON ] PR GPROP :NA "TELE
MAKE "WART RC
IF :WART = "M [MENU]
LABEL "NG
PR [LEIDER IST NAME NICHT VORHANDE]
TYPE [IRGENWELCHE TASTE ZUR MENU]]
IF :TAS = "M [MENU]
END

TO VOR
MAKE "ZL 0
MENU
END

TO ABSPEICHER
CT
PR [* * * * * * ADRESS BUCH * * * * * *]
REPEAT 4 [PR "]
PR [BITTE NAME DER DATEI EINGEBEN]
PR [NAME  ]
MAKE "DN FIRST RL
SAVE :DN
MENU
END
```

Informationen

Der **ST-Pascal**-Compiler von CCD ist schon seit einigen Wochen in einer überarbeiteten Version 1.04 verfügbar. Besitzer der alten Version erhalten sie samt deutschem Handbuch zum absoluten Selbstkostenpreis (8,– DM + Porto). Die neue Version bietet neben einer komfortableren Bedienung zusätzliche Grafikbefehle (Line-A Befehle) und einen neueren, schnelleren Linker. In Kürze soll ein ST Pascal+ erscheinen, worin auch die VDI- und AES-Befehle implementiert sind. Ferner soll diese Version ein Programm erzeugen, das in punkto Geschwindigkeit und Länge optimiert wird. Die neu implementierten Grafik-Befehle sollen wesentlich leichter anzusprechen sein als die eigentlichen GEM-Routinen. Die im Februar-Heft angesprochenen Kritikpunkte wären somit beseitigt.

ST-Pascal

Die Benutzung einer Festplatte oder ausreichend dimensionierten RAM-Disk ist bei diesem Compiler auch möglich. Dabei muß man lediglich das neue Laufwerk angeben und beachten, daß sich die Originaldiskette im Laufwerk befindet. Dadurch wird die Compilierzeit deutlich verkürzt.

Vier Programme zur besseren Ausnutzung des ST-Betriebssystems werden von Herrn Michael Gehret angeboten. Bei dem ersten Programm mit dem Namen **Coco** handelt es sich um einen File-Generator für DRI CP/M 68K / GEMDOS C-Compiler und Linker. Coco generiert Kommandos zur Erzeugung und Verwaltung aller Daten bei der C-Programmentwicklung. Wie beim UNIX cc-Kommando ist der Aufruf von Coco unabhängig von der Art der zu verarbeitenden Dateien.

Das zweite Programm ist ein GEM-Editor namens **GED**. Es bietet hohe Funktionalität und einfache Bedienung bei schneller Ausführung.

GED gibt es in zwei Versionen:

T für ATARI ST Standalone unter TOS

X für ATARI ST als intelligentes UNIX-Terminal, Teile des Editors sind auf den UNIX-Host ausgelagert, dessen Terminal-I/O-Last erheblich vermindert wird.

Eine Erleichterung für die C-Programmierung von AES- und VDI-Schnittstellen bietet **CAESVDI**. Der Gebrauch vieler AES- und VDI-Funktionen ist wegen der hohen Parameterzahl recht umständlich. CAESVDI löst diese Probleme und erhöht so die SE-Produktivität. Die Schnittstellen sind zwischen verschiedenen GEM-Umgebungen vollkommlen portabel.

Das letzte Programm **Sigh** decodiert die vom TOS Traphandler bei einem Systemabsturz geretteten Informationen. Es kann zur späteren Auswertung in eine Datei umgelenkt werden.

Hiermit teile ich Ihnen mit, daß ich einen Konvertierungsservice von Apple-CP/M Software in das Atari-Format anbiete. Die Konvertierungskosten betragen pro Disk incl. 3.5 Zoll Leerdiskette 40,– DM. Der Kunde schickt seine Apple-Diskette zu mir, Sie werden konvertiert; der Kunde erhält Originaldisk und konvertierte Disk zurück. Vorkasse oder Verrechnungsscheck.

Sönke Kastner, Wedel

Die Firma BASIS COMPUTER SYSTEME aus Münster bietet für den 520 ST und 260 ST eine RAM-Erweiterung auf 1 MByte. Die zusätzlichen Speicher-Bausteine wurden auf eine Extra-Platine untergebracht. Der Umbau erfolgt durch Auslöten dreier IC-Bausteine und Anlöten der Signale CAS und RAS. Die Platine ist für 265,– DM bei oben erwähnter Firma zu beziehen.

Dateisysteme

ST-Manager Kunden, Lager, Rechnung ist nun lieferbar. Es handelt sich hierbei um ein mächtiges Programmpaket, das die gesamte Verwaltungsarbeit einer Firma übernehmen kann. Informationen hierzu bei Fachhändlern oder bei ATARI.

Von RDS gibt es unter dem Namen 'Adress perfekt' eine neue, einfach bedienbare Adressdateiverwaltung. Sie läuft unter GEM und bietet einige interessante Eigenschaften. Dazu gehören verschiedene Ausdruckformate, variable Druckeranpassungen (Schriftartumschaltung, ...) und vor allem vielseitige Auswahlkriterien. So kann man in allen Feldern, die leider festgelegt sind, aber alle nötigen Daten enthalten, nach einer bestimmten Information suchen (beispielsweise nach Postleitzahlen von 5000–5999) und somit einen bestimmten Personenkreis auswählen.

Die so ausgewählten Daten können beispielsweise als Adressaufkleber gedruckt oder zur Weiterverarbeitung an Textverarbeitungsprogramme übergeben werden (RDS Software, 148 DM). Die Datensätze sind gegen unbefugte Benutzung durch ein Passwort, das beim Laden eines Datensatzes verlangt wird, geschützt.

Ein ähnliches, GEM-unterstütztes Programm, das aber momentan noch in englischer Sprache geliefert wird, heißt 'Hipposimple' und ist über ATARI (zum Preis von 149 DM) zu beziehen.

Ein Karteiverwaltungsprogramm wird von der Firma HOFACKER angeboten. Es dient zum Anlegen verschieden großer Dateikarten, welche nach bestimmten Kriterien durchsucht, angezeigt und ausgedruckt werden.

Einen vollen Erfolg verzeichnet micro team gmbh, 6070 Langen, Gartenstraße 102, Deutschland-Distributor der CANON Rechner mit den auf der micro computer '86 neu vorgestellten Softwarepaketen für Laserdrucker.

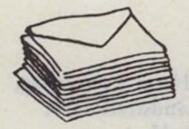
micro team, spezialisiert auf den Vertrieb und Support der CANON-Laserdrucker, zeigte **micrograf II**, eine komfortable Hardcopy-Routine für die Laserdrucker LBP-8 A1/A2. Vier unterschiedliche Druckformate, Positiv- oder Negativdarstellung, Positionierung und Text/Grafikmischung erweitern die Einsatzmöglichkeit des Laserdruckers erheblich.

microform erlaubt die Erstellung eigener Formulare auf dem Bildschirm. Vordrucke wie Rechnung, Lieferscheine oder Urkunden werden einmal erstellt und stehen dem Anwender jederzeit von der Diskette zur Verfügung. In den Druckerspeicher (128 K bzw. 1,1 Mb) geladen, können mit jeder Textverarbeitung Formular und Text in einem Arbeitsgang gedruckt werden. Schattierte Felder und unterschiedliche Schriftgrößen ersparen dem Anwender den Gang zur Druckerei und vermeiden unnötige Kosten bei Änderung der Firmendaten.

Anpassungen an nahezu jede Soft- und Hardware sind realisiert, seit kurzem stehen auch Druckertreiber für die Atari 520 Software 1st-Word, C-Text und SM-Text zur Verfügung; ein Atari-Grafiktreiberprogramm wurde angekündigt.

Für Schriftsetzer und Buchdrucker soll in Kürze das Paket **microdruck** für erhebliche Kostensenkung sorgen: Proportionaler Blocksatz, freie Spalteneinteilung mit Fußnotenverwaltung in Verbindung mit dem Laserdrucker ermöglichen Schriftsatz zu bisher nicht realisiert niedrigen Betriebskosten.

Als weitere Projekte wurden angekündigt: Schriftsatz mit Punktgrößeneingabe, die den seitenfüllenden Ausdruck eines Buchstabens ermöglichen, sowie hochauflösende Ausdrucke von Videoaufnahmen.



Ich möchte Sie und die Leser Ihrer Zeitschrift auf einen gravierenden Fehler im Basic Interpreter des Atari ST 520+ aufmerksam machen:

Werden bestimmte Zahlenwerte beliebigen Variablen zugewiesen, so kann dies zur Fehlermeldung "Function not yet done" oder sogar zum Systemabsturz führen.

Beispiele:

a = 75 geht problemlos
a = 75,5 ergibt Fehlermeldung
a = 75,7 erzeugt Totalabsturz
a = 75,8 ...wer weiß was dann passiert!

Wer noch nicht genug hat, kann ja mal a = 276E–12 probieren.

Sollte dieser Fehler bei der "allerneuesten" Version des Interpreters nicht mehr auftreten, bitte ich um kurze Nachricht.

Anfried Sauerborn, Haar

Antwort: Zunächst dachten wir Sie hätten eine uralte Version des ST-Basics. Nachdem wir allerdings Ihre Zeilen der neusten Basicversion zur Verarbeitung anboten, brach, als sich der Rechner mit zwei Bömbchen bedankte, in der Redaktion helle Freude aus. Leider haben Sie die neuste Version.

Unser Tip: Warten Sie auf das neue Basic.

Die Redaktion

Denkspiel "Supercode", und technische Frage

Mit großer Freude habe ich die Februar ST gelesen, in der auch ein Spiele-Listing abgedruckt ist.

Gleich habe ich mich vor meinen Atari 520 ST+ gesetzt, das Basicprogramm mit meiner Floppy SF 314 geladen, das Listing von Supercode eingetippt und das Ergebnis auf meinem RGB Monitor Thomson für Atari erwartet.

Leider mußte ich feststellen, daß das Programm nicht auf einem Farbmonitor läuft, hätte man das nicht im Text oder im Listing schreiben können, so hätte ich mir 2 Stunden Arbeit erspart. Vielleicht können Sie mir ein geändertes Listing zuschicken?

Ferner habe ich bei meinem Atari-Händler nach den Programmen gefragt, die kostenlos auf eigene Disketten kopiert werden dürfen. Mein Atari-Händler sagte mir, daß Ihr Bericht auf Seite 69/70 der ST Nr. 2, nicht der Wahrheit entspricht, Atari hätte nur für Besitzer des alten 520 ST Programme zum kopieren frei gegeben. Es läge keine andere Stellungnahme von Atari vor.

Eine technische Frage habe ich noch. Um mir das Ein- und Ausschalten jedes einzelnen Gerätes (520 ST+, SF 314, RGB Monitor und später eines Druckers) zu ersparen, möchte ich die Schalter an den Geräten auf "Ein" lassen und die Steckdosenleiste mit den Netzsteckern über einen Schalter bedienen. Ist dieses gefahrlos für den Computer möglich?

Vielleicht können Sie mir mit meinen 3 Problemen weiter helfen?

Zu Ihrer Zeitschrift möchte ich sagen, sie könnte zweimal im Monat erscheinen, dann wären die Wartezeiten nicht so lange auf das nächste Heft.

Für eine baldige Antwort wäre ich Ihnen dankbar.

Rüdiger Nürge, Celle

Antwort: Ihr Problem mit dem Basicprogramm auf einem Farbmonitor beruht auf der unterschiedlichen Koordinatenaufteilung zwischen dem monochromen und den zwei Farbmodi.

Bei der Anpassung auf die mittlere Auflösungsstufe müssen lediglich alle y-Koordinaten der Ein/Ausgabefunktionen halbiert werden, bei der niedrigen zusätzlich die x-Koordinaten. Das gleiche gilt für die Schriftgröße.

Zu Ihrer Frage bezüglich des Einschaltens der Geräte. In unserer Redaktion schalten wir seit geraumer Zeit grundsätzlich alle Geräte mit einem Schalter ein und aus, wobei noch keine Probleme auftraten. Natürlich können wir keine Garantie für eventuell auftretende Schäden übernehmen.

Seit vier Monaten besitze ich einen, inzwischen aufgerüsteten, ATARI 520 ST. Ich finde es toll, daß es endlich eine spezielle Zeitschrift für den ST-Anwender gibt. Für die vielfältigen Beiträge kann man euch nur ein Lob aussprechen. Besonders gefreut hätte mich der Druckertreiber (Heft 2, S. 49), da ich aus meiner früheren Computerzeit (C-64) noch einen NEC 8023B-N besitze. Nur leider fehlt mir ein Assembler um das Programm einzugeben.

Da ich bestimmt nicht der einzige Leser bin, der keinen Assembler besitzt, möchte ich folgenden Verbesserungsvorschlag anbringen:

Zu jedem Assembler-Programm (Utilities) ein entsprechendes Ladeprogramm in BASIC.

Eric Hofmann, CH-Bernhardzell

Antwort: Ihr Problem wurde in der ST-Redaktion auch viel diskutiert. Letztendlich haben wir beschlossen vorerst keine DATA-Zeilen abzudrucken, da man dann nicht nur Assembler-, sondern zum Beispiel auch Pascal- und C-Listings als Basic-Datas bringen müßte. Ferner würden seitenlange, nichtssagende Datazeilen bestimmt nicht zur Steigerung der Qualität unserer Fachzeitschrift beitragen. Damit Sie trotzdem Ihren Drucker weiterhin betreiben können, verweisen wir Sie auf unseren Diskettenservice.

Trotz unserer internen Entscheidung würde uns die Meinung weiterer Leser zum Thema Data-Zeilen interessieren. Bei entsprechend großer Nachfrage sind wir natürlich gerne bereit unsere Meinung darüber zu ändern.

Die Redaktion



Dementi

In unserer letzten Ausgabe ist uns leider der Fehler unterlaufen, die Programme DB Master One und 1st-Word fälschlicherweise als Public Domain Programme zu bezeichnen. Diese beiden Programme werden nur den Besitzern des alten 520 ST als Ersatz für die versprochenen GEM-Write und GEM-Paint kostenlos überlassen. Für alle anderen ST-Besitzer sind DB Master One und 1st-Word für je 99,– DM zu erwerben. Ferner haben wir uns nach zahlreichen Leserbriefen direkt bei ATARI erkundigt, ob ST-Händler die Public Domain Software kopieren lassen müssen. Die Antwort lautete, sie müssen es nicht, aber sie **dürfen** es. Wir bieten aber ab sofort einen Diskettenservice für Public Domain Software an. Die Namen der Programme, die zu erhalten sind, entnehmen Sie bitte unserem Angebot auf dieser Seite.

Vorschau

- Das Betriebssystem des ST
- Ausführliches Computerlexikon
- komfortable Druckeranpassung für 1st Word
- Resource Construction Set
- Fortsetzung der Kurse Pascal & GEM
- Hardwaretricks

Mit 5 Mark sind Sie dabei!

Die ST-Redaktion bietet für Sie ab sofort einen neuen Service an. Dieser wurde durch die große Anzahl von Leserbriefen angeregt. Für nur 5,– DM Unkostenbeitrag **pro Diskette** für das Kopieren und Bearbeiten, bieten wir folgende Public Domain Programme als Kopierservice an:

1. Diskette
   - Neochrome V.0.6: luxuriöses Malprogramm für alle Farbmonitorbesitzer
   - Dr. Doodle: Malprogramm für monochromen Monitor

2. Diskette
   - Joshua: Monitor und Diskmonitor
   - CP/M Emulator (CP/M 80)
   - Megaroids: Spiel für Monochrom- und Farbmonitor
   - Diskformatter: erweitert Diskkapazität auf 399 bzw. 809 kByte
   - Kartei-Kasten: luxuriöses Karteiprogramm unter GEM
   - 50/60 Hz-Umschaltung für Monitor im Autostart-Ordner

Für eine problemlose Bearbeitung müssen Sie folgendes beachten:

- Schicken Sie uns eine einseitig formatierte Diskette.
- außerdem einen ausreichend frankierten Rückumschlag, auf dem Ihre Adresse steht.
- Legen Sie 5,– DM als Scheck bei.

Die Bezugsadresse lautet: Uwe Bärtels · ST-Redaktion
Postfach 1131 · 6242 Kronberg

Impressum

ST-Computer

Herausgeber: Heim Fachverlag, Heidelberger Landstraße 194, 6100 Darmstadt 13, Telefon (0 61 51) 5 53 75

Redaktion: Uwe Bärtels (UB-Chefredakteur), Harald Schneider (HS), Marcelo Merino (MM), Harald Egel (HE)

Freie Mitarbeiter: Markus Nerding (MN),

Weitere Autoren: Bernhard Wurm (BW), Axel Quentin (AQ)

Titelseite: Klaus Ohlenschläger

Anzeigen: Anzeigenleiter H. Heim, Telefon (0 61 51) 5 53 75
Anzeigenpreise nach Preisliste Nr. 1 gültig ab 1.1.86

Erscheinungsweise: 11 x jährlich

Bezugspreis: Einzelheft DM 6,–. Jahresabonnement DM 60,– inklusive der gesetzlichen Mehrwertsteuer und den Zustellgebühren für 11 Ausgaben.

Bezugsmöglichkeiten: ATARI-Fachhändler, Zeitschriftenhandel, Kauf- und Warenhäuser oder direkt beim Verlag unter obiger Adresse.

Druck: Ferling Druck Darmstadt

Manuskripteinsendungen: Programmlistings, Bauanleitungen und Manuskripte werden von der Redaktion gerne angenommen. Sie müssen frei von Rechten Dritter sein. Mit ihrer Einsendung gibt der Verfasser die Zustimmung zum Abdruck und der Vervielfältigung auf Datenträgern im Heim Verlag. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte wird keine Haftung übernommen.

SOFTWARE FÜR ATARI ST

Produkt		VK-Preis
Adress-Perfect	**Adressverwaltung**	**148,--**
läuft unter GEM, suchen & sortieren nach allen Feldern. Schnittstelle zu beliebiger Textverarbeitung. SERIENBRIEFSCHREIBUNG		
Side-Click	**Terminplaner**	**198,--**
mit Taschenrechner, Kalender, Uhr Terminplanung und Überwachung usw.		
VT 100	**Terminal-Emulator**	**248,--**
unter GEM mit Upload/Download, Protokolldruck und voll VT 100 kompatibel		
C-Library	**Standard**	**128,--**
C-Library	**Extra**	**128,--**
C-Library	**Grafik**	**128,--**
C-Library	**GEM 1**	**128,--**
C-Library	**GEM 2**	**128,--**
C-Library	**Resource**	**128,--**
Die große Bibliothek für C-Programmierer näheres siehe INFO 1/86		
C-Library	**UNIX 1**	**158,--**
C-Library	**UNIX 2**	**158,--**
TBC-Compiler	**The Best C-Compiler**	**498,--**
inkl. Assembler, Linker & UNIX Librarys - Verarbeitet UNIX Sourcecode - Optimiert wie kein anderer!!! Softwareentwicklung auf ST - nur mit TBC		

Produkt		VK-Preis
TBC-LINT	**The Best C Lint**	**298,--**
LINT - Das Präparat gegen Typenkonflikte, schlechte Argumentübergabe, Variablenmüll, Strukturfehler usw. Gegenanzeige: Nicht anwenden bei Allergie gegen saubere Programmierung		
TBC-Development	**TBC-Compiler und LINT**	**748,--**
Profi-Monitor	**ST System-Monitor**	**298,--**
Direkt Assembler, Disassembler, Debugger, Hex-Ascii-Binär-Editiern		
Profi-Assembler	**ST Assembler**	**198,--**
Der preiswerte, komfortable ST Assembler		
Editor-Disk	**Icon, Font & Maus Editor**	**148,--**
läuft unter GEM, muß man einfach haben!		
ARST	**Archive Programm**	**98,--**
Für TBC und Entwicklungspaket!		
Disc-Monitor	**Disketten-Monitor**	**178,--**
Der bewährte, jetzt noch besser!		
Fastos Cartridge		**248,--**
DOPPELTE GESCHWINDIGKEIT bei allen Floppy-Arbeiten		
File-Copy +	**Fast File-Copy**	**98,--**
Wie oben, jedoch für Files. Wahnsinnig schnell!!!		
Drucker-Disk	**Drucker Utilitys**	**98,--**
Drucker-Treiber, Drucker-Spooler, Druck-Programm		

Fordern Sie unsere Info an!



Jakobstraße 8 a · 6096 Raunheim · ☎ (0 61 42) 4 31 42

GENERALVERTRETUNG SCHWEIZ:
SAMA Electronics
Bahnhofstraße 7
CH-5400 Baden

Alle Preise sind empfohlene Endverbraucherpreise incl. 14 % Mehrwertsteuer. Änderungen vorbehalten. Nähere Informationen erhalten Sie bei Ihrem ATARI-Händler oder direkt bei uns.

FORTH-SYSTEME ANGELIKA FLESCH

- **FORTH-System**

4 x FORTH

- **Level 1** – extrem schneller FORTH-Compiler 32 Bit **498,– DM**

Dieser Forth-Compiler ist extrem schnell (s. Testbericht ANTIC 12/85). Er bearbeitet ca. 100 000 Leerschleifen pro Sekunde, übersetzt ca. 20 KB-Source in 3 sec. incl. Linking und unterstützt Multitasking sowie Multiuser-Fähigkeiten.

Andere Software für den ATARI ST

- **H & D BASE** **348,– DM**

In FORTH geschriebenes dBase II kompatibles Daten- u. Dateiverwaltungs-System mit Ausnützen der ATARI-spezifischen Möglichkeiten (Windows, Maus etc.). Unterstützt den kompletten Speicherbereich. Volle Datei-Kompatibilität.

- **ST-Colouring-Book** **125,– DM**

Pictures für Neochrome (2 Discs)

Ausführliche Information bei

FORTH-SYSTEME ANGELIKA FLESCH
7820 Titisee/Neustadt Postfach 12 26 Tel. 0 76 51 / 16 65 + 33 04

Händler-Anfragen erwünscht.